PROGRAMAR

REVISTA PORTUGUESA DE PROGRAMAÇÃO • WWW.PORTUGAL-A-PROGRAMAR.PT

EDIÇÃO #51 - DEZEMBRO 2015

ISSN 1647-0710

TRAVESSIA
DE UMA ÁRVORE DE DIRETÓRIOS USANDO RECURSIVIDADE

ANÁLISE

MYSQL

A PROGRAMAR

VALIDAÇÃO DE FORMULÁRIOS EM JAVASCRIPT

APLICAÇÕES MOBILE COM O XAMARIN STUDIO

PLOTAGEM DE DADOS "EM TEMPO REAL" COM PYTHON USANDO MATPLOTLIB

XAMARIN 4 TEM TUDO QUE PRECISA PARA CRIAR ÓPTIMAS APLICAÇÕES MÓVEIS!

ELECTRÓNICA

AQUISIÇÃO DE DADOS VIA TCP/IP COM GENUINO (ARDUINO)

COLUNAS

CROSS-PLATFORM A SOMA DE TODAS AS MUDANÇAS **KERNEL PANIC**

NO CODE

AS NOVIDADES DA ATUALIZAÇÃO DE NOVEMBRO **WINDOWS 10**

A NOVA VERSÃO DO JAVASCRIPT **ECMASCRIPT 2015**

O NOVO MEMBRO DA FAMÍLIA! **RASPBERRY PI ZERO**

LISBON HAS BEEN HACKED **BETA-I HACKATHON**

E AINDA, IMPRESSORAS 3D, XAMARIN, HIGH TECH FASHION

EDITORIAL

EQUIPA PROGRAMAR

Coordenador
António Pedro Cunha Santos

Editor
António Pedro Cunha Santos

Design
Sérgio Alves
Twitter: @scorpion_blood

Ilustração
Sara Freixo

Redacção
André Melancia
André Silva
António Pedro Cunha Santos
Carlos Grilo
Esteban Solano
Fábio Pinho
Filipa Peres
José Viana
Nuno Silva
Patrício Domingues
Patrick Freitas
Rita Peres
Sara Silva
Tânia Valente
Vítor Carreira

Staff
António Pedro Cunha Santos
Rita Peres
Rui Gonçalves

Contacto
revistaprogramar@portugal-a-programar.org

Website
http://www.revista-programar.info

ISSN
1 647-071 0

Christmas_tree.cpp

```
#include "stdafx.h"
#include<stdio.h>
 void main()
  { int rows,starNo,spaceNo;  //Perguntar e definir metas próprias
     printf("Enter Rows:\n"); scanf("%d",&rows);
          // Definir o algoritmo
     for(int i=1;i<=rows;i++) {starNo=i*2-1; spaceNo=i+rows-starNo;
      for(int j=0;j<spaceNo;j++) {printf("%c",' ');}
      for(int k=0;k<starNo;k++)  {printf("%c",'*');}
     printf("\n"); }
   //Continuar a apostar no algoritmo
    for(int l=0;l<3;l++){ for(int m=0;m<(rows*2+1)/2;m++) {printf("%c",' ');}
      printf("%c\n",'*'); } printf("\n ");
          /*
```

Eis-nos chegados à última edição do ano 2015, com a nossa equipa a deixar aos leitores mais uma edição em jeito de prenda no sapatinho. É tempo de os geeks descansarem e darem atenção à família (*e que se note que pra mim, o meu portátil tem toda a legitimidade de receber o seu próprio presente uma vez que até partilhamos vários jantares ao longo do ano…).*

Nesta época sentimentalista em que recordamos o ano que agora termina e em que quase todos desejamos criar o algoritmo que nos permita encontrar as prendas perfeitas, quisemos tornar esta edição numa edição de todos e para todos. Para os programadores e para os designers. Para os fanáticos da informática e para os simplesmente simpatizantes.

Em suma, para os geeks e para os não geeks. Nesta edição há artigos para todos os gostos e feitios, sem nunca esquecer quem somos e para quem escrevemos.

E escrevemos para ti, caro leitor, quer nos acompanhes há várias edições, quer seja esta a primeira que lês. Esta edição é dedicada a cada um de vós.

Que 2016 seja um ano de triunfos, que tenham inspiração, que as vossas linhas de código se reinventem e as vossas compilações não falhem, que 2016 vos traga ideias e oportunidades para as por em prática!

Porque nós, programadores, somos “gente de ideias” e não desistimos!

Por isso e por muito mais… ***PROGRAMAR***… ontem, hoje e amanhã! Até à próxima edição.

```
          */

          printf("\nFELIZ NATAL e um BRINDE a 2016");
          scanf("%d",&rows);
 }
```

Rita Peres

ÍNDICE

TEMA DE CAPA

A PROGRAMAR

ELECTRÓNICA

COLUNAS

ANÁLISES

NO CODE

EVENTOS

- Shift APPens (19 a 21 de Fevereiro de 2016)
- SINFO (23 a 28 de Fevereiro de 2016)

Para mais informações/eventos: http://bit.ly/PAP_Eventos. Divulga os teus eventos para o email eventos@portugal-a-programar.pt

#PGGD 29

No passado dia 26 de Novembro, teve lugar mais um encontro do Portugal Girl Geek Dinner em Lisboa. Desta vez o espaço foi gentilmente cedido pela Main Hub - Innovation, Incubation & Development, que apoiou o projecto.

Estivemos na última sessão do grupo e adorámos!

Como é costume, o bom ambiente reinou na reunião, sendo que desta vez as oradoras convidadas foram a Ana Rita Queiroz, que nos trouxe uma apresentação sobre *"Os desafios do Marketing e da Comunicação no sector das TIs em Portugal"* e a Sofia Azevedo, que nos falou acerca de *"(Un)process or the truth about functional analysis"*, onde acentuou as verdades e os mitos sobre a análise funcional de um projecto.

As apresentações decorreram em ritmo animado sendo que reinou a partilha de experiências e conhecimentos. Já vos falámos das várias sessões das Portugal Girl Geek Dinners (fundadas em 2010 pela Vânia Gonçalves) e garantimos que vêm a adquirir cada vez mais adeptas (e adeptos) em Portugal.

No final da sessão, foram ainda sorteadas entre os participantes, três licenças e-learning da Pluralsight que foram gentilmente oferecidas a esta causa.

Deixamos um agradecimento especial à Mónica Rodrigues, Cláudia Carvalho (*organização*) e à Carla Quintino (*Main Hub*). Mais uma vez reiteramos que apesar de o *#PGGD29* ter sido em Lisboa, várias são as cidades portuguesas que se inserem neste projecto!

Aqui na Programar vamos continuar a acompanhar de perto, sendo que desafiamos todas as Geek Girls (e acompanhantes) a frequentarem as sessões!

Mais informações em: http://portugalgirlgeekdinners.com/blog/

WUD2015 - Dia Mundial da Usabilidade

Teve lugar no passado dia 12 de Novembro, assinalando a data, a conferência do Dia Mundial da Usabilidade. O evento aconteceu no auditório do ISLA de Gaia, com a parceria entre a instituição anfitriã e a WeMake, tendo presidido à sessão de abertura a Directora da Escola Superior de Tecnologia, a Drª Ana Paula Pinto e o Dr. Filipe Morais, CEO da WeMake.

No painel foram abordados temas como desafios e soluções de usabilidade, investigação através do desenvolvimento, ludificação e usabilidade, bem como Data Driven. Após o tradicional coffee break foi a vez de usabilidade na Web e Mobile, novas formas de interactividade em plataformas de entretenimento, usabilidade para a utilidade e usabilidade no Marketing.

Com um auditório recheado, trocaram-se ideias , trazendo a destaque a experiências dos vários oradores e respectivas empresas, bem como dos participantes no evento.

Ficam algumas imagens que marcaram a data.

TEMA DE CAPA

Travessia de uma árvore de diretórios usando recursividade

Travessia de uma árvore de diretórios usando recursividade

Introdução

O diretório é um elemento familiar nos sistemas de ficheiros, sendo empregue para organizar o armazenamento de dados em suporte persistente como, por exemplo, discos rígidos. Uma operação relativamente comum num sistema de ficheiros é a travessia da árvore de diretórios, através da qual são visitados todos os subdiretórios e ficheiros. Este artigo foca o uso de recursividade e das funções stat e readdir para a travessia de uma árvore de diretório recorrendo à metodologia denominada de *busca em profundidade* (*depth-first search* na designação anglo-saxónica) (Knuth, 1968) (Wikipedia, 2015). Embora o artigo assente no ambiente Linux e na linguagem C, a metodologia utilizada pode ser empregue, com algumas adaptações, noutras plataformas e com outras linguagens de programação.

Diretórios, caminhos absolutos e caminhos relativos

Um diretório pode conter ficheiros, outros diretórios e assim sucessivamente, resultando numa estrutura em árvore. O ponto de entrada da árvore é designado por diretório **raiz**, sendo representado pela barra **/** (*slash bar* na designação anglo-saxónica) em sistemas operativos UNIX e derivados. A barra / é ainda empregue como separador na representação de caminhos dos sistemas de ficheiros que identifiquem um determinado recurso, seja um diretório ou um ficheiro. No que respeita a caminhos, esses podem ser representados de forma absoluta ou relativa. A representação absoluta requer que o caminho do recurso que se pretende identificar seja especificado desde a raiz, iniciando-se pela barra /. Por exemplo, o caminho absoluto /bin/ls identifica a localização tradicional do comando ls num sistema UNIX. Por sua vez e como o nome sugere, um caminho relativo é especificado em relação ao diretório corrente, não se iniciando, portanto, com a barra /. Por exemplo, o caminho programar/artigo.doc refere-se ao ficheiro artigo.doc que se encontra no diretório programar que existe no diretório corrente. A Listagem 1 mostra o primeiro nível da árvore de diretórios definida pelo sistema de ficheiros de um sistema Linux.

```
/
├── bin
├── boot
├── dev
├── etc
├── home
├── lib
├── lost+found
├── media
├── mnt
├── opt
├── proc
├── root
├── run
├── sbin
├── srv
├── sys
├── tmp
├── usr
└── var
```

Listagem 1: árvore de diretórios de um sistema Linux a partir do diretório / (saída do tree: tree -L 1 -d /)

As funções da família *stat*

As funções da família stat – stat, fstat e lstat – permitem o acesso aos metadados de uma entrada do sistema de ficheiros, seja um diretório, um ficheiro, uma ligação simbólica ou um outro recurso. A designação metadados abrange os elementos caracterizadores da entrada, tal como a data de criação, as permissões, a identificação do dono e grupo do recurso, entre outros elementos. Para o efeito, as funções da família stat recorrem à estrutura struct stat para representação dos metadados. A struct stat tem 13 campos, apresentados na Listagem 2 que foi diretamente extraída da saída do comando man 2 stat. Note-se que algumas versões de UNIX podem ter campos adicionais na struct stat, sendo que o uso desses campos extra num programa limita necessariamente a portabilidade do código.

```
struct stat {
dev_t     st_dev;/* ID of device containing file */
ino_t     st_ino;/* inode number */
mode_t    st_mode;/* protection */
nlink_t   st_nlink;/* number of hard links */
uid_t     st_uid;/* user ID of owner */
gid_t     st_gid;/* group ID of owner */
dev_t     st_rdev;/* device ID (if special file) */
off_t     st_size;/* total size, in bytes */
blksize_t st_blksize;/* blocksize for filesystem I/
O */
blkcnt_t  st_blocks;/* num of 512B blocks allocated
*/
time_t    st_atime;/* time of last access */
time_t    st_mtime;/* time of last modification */
time_t    st_ctime;/* time of last status change */
};
```

Listagem 2: estrutura struct stat (fonte: man 2 stat)

A Listagem 3 apresenta o código da aplicação metadados_dir_corrente que exemplifica o uso da função stat, mostrando na saída padrão o valor dos 13 campos de metadados da struct stat para a entrada "." (ponto). A entrada "." é uma entrada especial que representa o diretório corrente. Outra entrada especial é o ".." (ponto ponto) que representa o diretório pai ou diretório ascendente. Tanto o diretório "." e o diretório *pai* ".." existem em todos os diretórios, pelo que as entradas "." e ".." são sempre válidas (no caso do diretório raiz /, a entrada ".." aponta para o próprio diretório raiz). A Listagem 4 mostra a saída resultante da execução da aplicação metadados_dir_corrente. Obviamente, a saída varia consoante o diretório corrente onde é executada a aplicação.

Em termos de código, para além da chamada à função stat, a aplicação metadados_dir_corrente implementa a função tipo_file_str que recebendo o campo st_mode da struct stat devolve uma cadeia de caracteres indicando se a entrada corresponde a um ficheiro regular, diretório, atalho (*symbolic link*) ou "outro". Para determinar o tipo de entrada, a função tipo_file_str() faz uso das *macros* do sistema S_IS_REG(), S_IS_DIR() e S_IS_LNK() indicando como parâmetro o campo st_mode da entrada em análise. Importa notar que para além do tipo de entrada, o campo st_mode contém, a informação referente às permissões de acesso do recurso – leitura, escrita, execução – para o dono, o grupo e o *outros*. Finalmente, os campos do tipo time_t, isto é, st_atime, st_ctime e st_mtime são convertidos para uma representação dia/hora através da função ctime_r que corresponde à versão *reentrante* da função ctime. Mais detalhes podem ser encontrados na respetiva página do manual (*man ctime)*.

Conforme referido anteriormente, a família de funções stat tem, para além da função stat propriamente dita, as funções fstat e lstat. Na função fstat, o ficheiro do qual se pretende os respetivos metadados é indicado por um descritor de ficheiro, descritor esse previamente obtido através da função open. Por sua vez, a função lstat comporta-se de forma similar à função stat, com uma exceção: quando é indicada uma ligação simbólica, o lstat devolve os metadados da ligação simbólica e não os metadados para a qual aponta a ligação simbólica. Mais adiante neste artigo, far-se-á uso da função lstat na implementação recursiva da travessia de uma árvore de diretórios.

```
#include <…>
char *tipo_file_str(mode_t st_mode){
      if( S_ISREG(st_mode) ){
            return "ficheiro normal";
      }else if( S_ISDIR(st_mode) ){
            return "Diretório";
      }else if( S_ISLNK(st_mode) ){
            return "Atalho";
      }
      /* ainda aqui? */
      return "outro";
}

void mostra_struct_stat(const struct stat *buf_ptr)
{
      char dia_hora_str[128];
      printf("st_dev = %u.%u\n",
            major(buf_ptr->st_dev),
            minor(buf_ptr->st_dev));
      printf("st_ino = %ld\n", buf_ptr->st_ino);
    printf("st_mode = %x (%s)\n", buf_ptr->st_mode,
            tipo_file_str(buf_ptr->st_mode) );
      printf("st_nlink = %d\n", buf_ptr->st_nlink);
      printf("st_uid = %d\n", buf_ptr->st_uid);
      printf("st_gid = %d\n", buf_ptr->st_gid);
    printf("st_rdev= %d\n", (int)buf_ptr->st_rdev);
      printf("st_size= %lu\n", buf_ptr->st_size);
      printf("st_blksize= %ld\n",
                  buf_ptr->st_blksize);
      printf("st_blocks=%ld (blocos de 512B)\n",
                  buf_ptr->st_blocks);
      /* ctime devolve string com '\n' no final */
      printf("st_atime (ultimo acesso)=%s",
      ctime_r(&(buf_ptr->st_atime),dia_hora_str));
      printf("st_mtime (ultima modificação)=%s",
       ctime_r(&(buf_ptr->st_mtime),dia_hora_str));
      printf("st_ctime (ultima alter. estado)=%s",
       ctime_r(&(buf_ptr->st_ctime),dia_hora_str));
}

int main(void){
      char *nome = ".";/* diretório corrente */
      int ret_stat;
      struct stat stat_buf;
      ret_stat = stat(nome, &stat_buf);
      if( ret_stat == -1 ){
            fprintf(stderr,"Erro: chamada stat"
            " para recurso '%s' - %s\n",
            nome, strerror(errno));
            exit(1);
      }
      mostra_struct_stat(&stat_buf);
      return 0;
}
```

Listagem 3: aplicação mostra_metadados_dir_corrente

```
st_dev = 8.17
st_ino = 3662149
st_mode = 41fd (Diretório)
st_nlink = 2
st_uid = 1000
st_gid = 1000
st_rdev= 0
st_size= 4096
st_blksize= 4096
st_blocks=8 (blocos de 512B)
st_atime (ultimo acesso)=Sat Oct 31 01:04:05 2015
st_mtime (ultima modificação)=Sat Oct 31 01:10:16
                                                2015
st_ctime (ultima alter. estado)=Sat Oct 31 01:10:16
                                                2015
```

Listagem 4: saída produzida pela execução de metadados_dir_corrente

Acesso ao conteúdo de um diretório

Em termos de programação, o acesso ao conteúdo de um diretório – ficheiros, diretórios, atalhos, etc. – apresenta algumas semelhanças com a leitura de um ficheiro, sendo necessário efetuar a abertura do diretório antes que se possa aceder ao conteúdo e fechar o diretório quando se terminou o respetivo uso. Para o efeito existem as funções opendir (abertura) e closedir (fecho). O acesso ao conteúdo é feito através de chamadas sucessivas à função readdir que devolve uma entrada por chamada. Os protótipos das funções necessárias para iterar o conteúdo de um diretório estão indicados na Listagem 5.

No acesso ao conteúdo de um dado diretório, a primeira operação consiste na abertura do diretório através da função opendir que, caso seja bem sucedida, devolve o endereço de memória de um descritor de diretório do tipo de dados DIR.

```
#include <sys/types.h>
#include <dirent.h>

DIR *opendir(const char *name);

struct dirent *readdir(DIR *dirp);
int readdir_r(DIR *dirp, struct dirent *entry,
                  struct dirent **result);
```

```
int closedir(DIR *dirp);
```

Listagem 5: protótipos das funções opendir, readdir e closedir (fonte: man opendir e man readdir)

O descritor devolvido pela função opendir é empregue para iterar o conteúdo do diretório através da função readdir. Esta função deve ser chamada em ciclo, devolvendo para cada chamada uma entrada do diretório. A função readdir assinala o término da iteração através da devolução do valor NULL, indicando que já foram iteradas todas as entradas do diretório. O passo final corresponde ao encerrar do diretório, efetuando-se através da chamada à função closedir. De seguida, analisam-se separadamente cada uma das etapas para acesso aos conteúdos de um diretório: abertura, leitura e encerramento do diretório.

Obtenção do descritor – opendir

A obtenção de um descritor do tipo DIR para um determinado diretório efetua-se através da função opendir. A função recebe como único parâmetro o caminho (absoluto ou relativo) do diretório para o qual se pretende receber um descritor. Caso a operação seja bem sucedida, a função devolve o endereço de uma estrutura do tipo DIR. Essa estrutura contém, entre outros elementos, um descritor para o diretório. Por questões de simplicidade, este artigo emprega a designação *descritor de diretório* para designar a estrutura do tipo DIR. Caso ocorra um erro na tentativa de abertura do diretório (por exemplo, o diretório especificado não existe), a função opendir devolve NULL, sendo atribuído à variável de erro errno o respetivo código numérico do erro. Dado que os códigos numéricos são pouco descritivos para os operadores humanos, é possível obter uma frase descritiva do código de erro indicado pelo valor da variável errno através da função strerror. A Listagem 6 apresenta o código do programa mostra_dir_corrente que faz uso da função strerror para tornar as mensagens de erro mais compreensíveis. Mais informações sobre a variável errno e a função strerror estão disponíveis nas respetivas páginas de manual eletrónico (man errno e man strerror).

Iteração do diretório – readdir

Obtido um descritor para o diretório, passa-se à fase de leitura das entradas existentes no diretório. Para o efeito faz-se uso da função readdir que atua de forma similar a um operador de iteração: sempre que é chamada, devolve uma entrada diferente do diretório até que estejam esgotadas todas as entradas. Deste modo, e caso se pretenda iterar todo o conteúdo do diretório, a função readdir deve ser chamada em ciclo até que seja alcançada a última entrada do diretório.

Como parâmetro, a função readdir recebe o descritor de diretório anteriormente obtido através da função opendir. Caso a chamada seja bem sucedida, a função readdir devolve o endereço de uma estrutura do tipo struct dirent associada à entrada corrente. Ao invés, se já tiverem sido percorridas todas as entradas do diretório, a função readdir devolve NULL, devolvendo ainda NULL na ocorrência de um erro, por exemplo, caso o descritor de diretório indicado seja inválido. Deste modo, torna-se necessário distinguir entre o término do diretório (situação normal) e a ocorrência de um erro (situação anormal). A distinção é feita através do valor da variável de erro errno que fica com um valor não nulo caso tenha ocorrido um erro na execução da função readdir. O código da função mostra_dir (Listagem 6) exemplifica o uso da variável errno para distinguir entre a normalidade do término do diretório e a anormalidade que é a ocorrência de erro na execução da função readdir.

De acordo com a norma POSIX é sempre garantido que a struct dirent tenha o campo d_name, que como o nome sugere, contém o nome da entrada. No Linux, a struct dirent contém mais alguns campos, nomeadamente o campo d_type que permite determinar o tipo da entrada (diretório, ficheiro regular, ligação simbólica, etc.). Contudo, caso se pretenda manter a portabilidade do código para ambientes POSIX, apenas deve ser empregue o campo d_name. Este artigo segue esta abordagem, fazendo uso da função stat para aceder aos metadados de uma entrada devolvido via struct dirent pela função readdir. A struct dirent disponibilizada em sistemas Linux é apresentada na Listagem 7.

```
#include <stdio.h>
#include <sys/types.h>
#include <dirent.h>
#include <errno.h>
#include <string.h>
#include <stdlib.h>

/*
 * Mostra as entradas do diretório 'nome_dir'.
 * Devolve 0 se tudo ok, -1 em caso de erro.
 */
int mostra_dir(const char *nome_dir){
    DIR *dir_d;
    int finito, n_entradas;
    struct dirent *dir_entry;
    dir_d = opendir(nome_dir);
    if( dir_d == NULL ){
      fprintf(stderr,"erro:"
      "impossível abrir DIR '%s' - %s\n",
            nome_dir, strerror(errno));
      return -1;
    }
    n_entradas = 0;
    finito = 0;
    do{
      dir_entry = readdir(dir_d);
      if( dir_entry == NULL){
            if(errno){
                fprintf(stderr,"erro:readdir"
                  (entrada %d)\n",n_entradas);
                closedir(dir_d);
                return -1;
            }else{
                printf("Iteração de DIR '%s' "
                  "terminada (%d entradas)\n",
                    nome_dir, n_entradas);
                finito = 1;
            }
      }else{
            printf("entrada: '%s'\n",
                  dir_entry->d_name);
```

```
            n_entradas++;
      }
    }while(finito == 0);
    if( closedir(dir_d) == -1 ){
      fprintf(stderr,"erro: impossível fechar "
      "DIR '%s' - %s\n", nome_dir, strerror
                                        (errno));
      return -1;
      }
    printf("DIR '%s': %d entradas\n",
                  nome_dir, n_entradas);
    return 0;
}
int main(void){
    char *nome_diretorio = "."; /* dir. corrente */
    int ret = mostra_dir(nome_diretorio);
    return ret;
}
```

Listagem 6: programa mostra_dir_corrente

```
struct dirent {
  ino_t    d_ino;       /* inode number */
  off_t    d_off;       /* not an offset */
  unsigned short d_reclen    /* length of record */
  unsigned char  d_type; /* type of file;
           not supported by all filesystem types */
  char           d_name[256]; /* filename */
};
```

Listagem 7: struct dirent em sistemas Linux (fonte: man readdir)

Encerramento

O encerramento de um descritor DIR é realizado através da chamada à função closedir. Essa operação é necessária para garantir que os recursos associados ao descritor são convenientemente libertados. A função closedir recebe o descritor do tipo DIR que se pretende encerrar. Caso o encerramento seja bem sucedido, a função devolve 0; caso contrário, devolve -1, atribuindo à variável errno um código numérico descritivo do erro. Na função mostra_dir (Listagem 7) exemplifica-se a verificação da eventual ocorrência de erros na sequência do fecho do diretório.

Uso combinado das funções readdir e stat

A travessia de uma árvore de diretórios requer a capacidade de identificar os tipos de cada entrada existente num diretório. De facto, torna-se necessário identificar as entradas do tipo diretório por forma a que esses subdiretórios possam eles também ser visitados. Para o efeito, torna-se necessário o uso combinado das funções readdir (convenientemente enquadrada com opendir e closedir) e stat. A Listagem 8 apresenta o código da aplicação mostra_dir_stat que faz uso das funções readdir e stat para listar o conteúdo do diretório cujo nome (absoluto ou relativo) é indicado à aplicação através do único argumento a linha de comando. A aplicação lista o conteúdo do diretório indicado, mostrando o tipo – diretório, ficheiro regular, ligação simbólica – de cada entrada. Assim, para cada entrada do diretório devolvida pela função readdir é chamada a função stat usando como parâmetro o campo d_name da struct dirent devolvido pela função readdir. Seguidamente, o campo st_mode da struct stat preenchido pela função stat é passado à função devolve_tipo_entrada que devolve uma cadeia de caracteres descritiva do tipo de entrada, tais como "*DIR*" para diretório, "*ligação simbólica*" para uma ligação simbólica, "*ficheiro*" para um ficheiro regular, etc. Importa notar que a determinação do tipo de entrada é feita aplicando-se uma operação de AND binário entre o valor st_mode e a máscara S_IFMT, comparando-se o resultado dessa operação com as constantes S_IFDIR (diretório), S_IFIFO (*pipe FIFO*), S_IFLNK (ligação simbólica), S_IFREG (ficheiro regular) e S_IFSOCK (*socket*). Tanto a máscara S_IFMT, bem como as constantes S_IFIFO, S_IFLNK, S_IFREG e S_IFSOCK são disponibilizadas quando se faz uso dos ficheiros de cabeçalho (.h) associados à função stat. Esta informação e mais alguns detalhes constam da página do manual eletrónico da função stat, acessível através de man stat.

```
#include <…>
/* Devolve string descritiva da entrada st_mode. */
char *devolve_tipo_entrada(mode_t st_mode){
    switch( st_mode & S_IFMT ){
      case S_IFDIR:
            return "DIR";
      case S_IFIFO:
            return "FIFO/pipe";
      case S_IFLNK:
            return "ligação simbólica";
      case S_IFREG:
            return "ficheiro";
      case S_IFSOCK:
            return "socket";
      default:
            return "outro";
    }
    return NULL;    /* Nunca alcançado */
}
/* Mostra as entradas do diretório 'nome_dir'
 * Devolve 0 se tudo ok, -1 em caso de erro */
int mostra_dir(const char *nome_dir){
    DIR *dir_d;
    int finito, n_entradas;
    struct dirent *dir_entry;
    char path[PATH_MAX];
    size_t path_len = sizeof(path);
    dir_d = opendir(nome_dir);
    if( dir_d == NULL ){
      fprintf(stderr,"erro: impossível abrir"
      "DIR '%s' - %s\n",nome_dir, strerror(errno));
      return -1;
    }
    n_entradas = 0; finito = 0;
    do{
      dir_entry = readdir(dir_d);
      if( dir_entry == NULL){
          if(errno){
            fprintf(stderr,"erro: readdir"
                 "(entrada %d)\n", n_entradas);
               closedir(dir_d);
               return -1;
          }
          printf("Iteração de DIR '%s' terminada "
            "(%d entradas)\n", nome_dir,
                                      n_entradas);
          finito = 1;
      }else{
          struct stat stat_buf;
          snprintf(path,path_len, "%s/%s",
            nome_dir,dir_entry->d_name);
          if( stat(path,&stat_buf) == -1 ){
```

```
            fprintf(stderr,"impossível stat"
                " '%s':%s\n", dir_entry->d_name,
                 strerror(errno));
            return -1;
          }
          n_entradas++;
          printf("[%03d]'%s' (%s)\n",
            n_entradas, dir_entry->d_name,
             devolve_tipo_entrada
                              (stat_buf.st_mode));
      }
    }while(finito==0);
    if( closedir(dir_d) == -1 ){
      fprintf(stderr,"erro: impossível fechar"
       "DIR '%s' - %s\n", nome_dir, strerror
                                          (errno));
      return -1;
    }
  printf("-----------------------------------\n");
    printf("DIR '%s': %d entradas\n",
      nome_dir, n_entradas);
    return 0;
}
int main(int argc, char *argv[]){
    if( argc != 2) {
      fprintf(stderr,"usage: %s <path_of_dir>\n",
            argv[0]);
    exit(EXIT_FAILURE);
    }
    char *nome_dir_ptr = argv[1];
    mostra_dir(nome_dir_ptr);
    return 0;
}
```

Listagem 8: programa mostra_dir_stat

Travessia de uma árvore de diretórios

Analisando-se uma árvore de diretório, facilmente se identifica um padrão repetitivo: uma árvore é composta por subárvores. De facto, um subdiretório pode ser considerado a raiz de uma subárvore do diretório que contém o subdiretório e assim sucessivamente. Deste modo, a travessia de um diretório e de todos os seus subdiretórios pode ser decomposta noutras operações de travessia, uma travessia por cada subdiretório. Na informática, este tipo de operação que se subdivide em operações do mesmo tipo em sub-conjuntos do conjunto original de dados presta-se ao uso de recursividade. De uma forma aproximada, pode dizer-se que em termos de programação programáticos, a recursividade consiste no chamar de uma função por ela própria por forma a tratar um subconjunto dos dados originais. O chamar da função por ela própria origina uma nova instância da função. No caso da travessia de uma árvore de diretórios, cada nova instância da função de travessia recebe um subdiretório do diretório que está a ser processado pela função chamante, invocando ela própria uma nova instância para cada subdiretório que tenha e assim sucessivamente.

Embora a descrição seja relativamente complexa, em termos de código, a operacionalização é simples: sempre que se deteta uma entrada do tipo diretório no diretório corrente, cria-se uma nova instância da função. Essa nova instância (i.e., chamada recursiva) recebe como parâmetro na chamada à função o caminho correspondente ao diretório corrente acrescido do nome do diretório a ser processado pela nova instância. Por exemplo, caso a função se encontre a processar o diretório dir_A/dir_B/dir_C e se torne necessário processar o subdiretório dir_D, a função é chamada com o caminho dir_A/dir_B/dir_C/dir_D. Quando termina a travessia desse subdiretório (e dos eventuais subsubdiretórios, etc.), a nova instância termina, regressando a execução ao diretório corrente para processamento da entrada seguinte e assim sucessivamente.

A Listagem 9 apresenta o código da função dir_recursivo que efetua a travessia uma árvore de diretórios e subdiretórios. O primeiro parâmetro – nome_dir – corresponde ao caminho do diretório a percorrer. Por sua vez, o segundo parâmetro – profundidade – indica a profundidade do diretório em análise. A profundidade mede o número de subdiretórios a que se encontra o diretório corrente em relação ao diretório original. Por exemplo, se a função for chamada com o diretório original / (raiz) num sistema Linux, o subdiretório /usr terá profundidade 1 e o subsubdiretório /usr/bin terá profundidade 2 e assim sucessivamente. Para o efeito, considera-se que o diretório original tem profundidade 0 (zero), sendo esse o valor que deve ser passado na primeira chamada à função dir_recursivo. Depois, para cada chamada recursiva, é passado o valor de profundidade + 1, refletindo o facto de que a nova instância da função processa um novo nível de subdiretório.

Na travessia recursiva de uma árvore de diretórios é necessário ainda ter em conta duas situações particulares. A primeira está relacionada com os diretórios “.” (ponto) e “..” (ponto ponto), que conforme anteriormente foi visto, correspondem respetivamente ao diretório corrente e ao diretório *pai*. Dadas as especificidades desses diretórios, torna-se necessário ignorá-los em termos de travessia recursiva. De facto, caso se considerasse o diretório “.” para efeitos de travessia, chamando recursivamente a função dir_recursivo com “.”, cair-se-ia num ciclo sem fim, preso no diretório corrente. Na prática, a aplicação terminaria devido a um excessivo nível de recursividade materializado com um transbordo de pilha ou *stack overflow* na designação anglo-saxónica (Chris Anley, 2007). Similarmente, o diretório “..” deve ser ignorado, pois levaria o controlo de execução para o diretório ascendente, impedindo o normal processamento do diretório corrente e respetivos subdiretórios. O ignorar dos diretórios “.” e “..” para efeitos de travessia consegue-se comparando o nome do novo diretório a visitar com “.” e “..”, comparação efetuada através da função strcmp.

A segunda situação a acautelar está relacionada com ligações simbólicas. Assim, caso seja detetada uma ligação simbólica para um diretório, essa mesma deve ser ignorada, caso contrário corre-se o risco de se saltar para outra árvore de diretório, interrompendo a travessia pretendida. A necessidade de deteção de uma ligação simbólica requer o uso da função lstat que, conforme anteriormente referido, no caso de uma ligação simbólica devolve os metadados referentes à ligação simbólica e não ao recurso (diretório/ficheiro) aponta-

do pela ligação simbólica. Desta forma, torna-se possível detetar uma ligação simbólica.

O programa dir_recursivo mostrado na Listagem 9 efetua a travessia de todos os diretórios, ficheiros e subdiretórios de uma árvore de diretório. A raiz da árvore de diretórios deve ser indicada como o único parâmetro da linha de comandos do programa. Como é óbvio, apenas são mostrados os conteúdos para os quais o utilizador que executa a aplicação tem permissões de acesso.

```
#include <...>
/* Devolve string descritiva da entrada
 * indicada por st_mode. Caso o tipo de entrada não
 * seja reconhecido, a função devolve "outro". */
char *devolve_tipo_entrada(mode_t st_mode){
    switch( st_mode & S_IFMT ){
      case S_IFDIR:
            return ("DIR");
      case S_IFIFO:
            return ("FIFO/pipe");
      case S_IFLNK:
            return ("ligação simbólica");
      case S_IFREG:
            return ("ficheiro");
      case S_IFSOCK:
            return ("socket");
      default:
            return ("outro");
    }
    /* Nunca alcançado */
    return NULL;
}
/* Função recursiva que percorre a árvore de
 * diretórios definida por 'nome_dir'.
'profundidade'
 * indica o nível de profundidade de nome_dir
 * relativamente à raiz da árvore de diretórios
 * em análise.
 * Devolve -1 em caso de erro, 0 se tudo OK.
 */
int dir_recursivo(char *nome_dir, int profundidade)
{
    DIR *dir_d;
    int finito, n_entradas;
    struct dirent *dir_entry;
    dir_d = opendir(nome_dir);
    if( dir_d == NULL ){
      fprintf(stderr,"erro: impossível abrir DIR "
            "'%s' - %s\n",nome_dir, strerror
                                        (errno));
      return -1;
    }
    n_entradas = 0;
    finito = 0;
    do{
      errno = 0;
      dir_entry = readdir(dir_d);
      if( dir_entry == NULL ){
          if(errno){
            fprintf(stderr,"erro: readdir "
            "(entrada %d)\n",n_entradas);
            closedir(dir_d);
            return -1;
          }else{
            printf("(DIR='%s',P=%d) "
               "DIR terminada (%d entradas)\n",
               nome_dir,profundidade,n_entradas);
            printf("-----------------------\n");
            finito = 1;
          }
      }else{
          char entrada[PATH_MAX]; /* path p/
                                          d_name*/
          snprintf(entrada,sizeof(entrada),"%s/%s",
                    nome_dir, dir_entry->d_name);
          struct stat stat_buf;
          if( lstat(entrada,&stat_buf) == -1 ){
            fprintf(stderr,"erro stat '%s':%s\n",
                        entrada, strerror(errno));
            continue;
          }
          n_entradas++;
          if( S_ISLNK(stat_buf.st_mode) == 1 ){
            fprintf(stderr,"Ignorado link %s\n",
                            entrada);
            continue;
          }
          if( S_ISDIR(stat_buf.st_mode) ){
            if( strcmp(".",dir_entry->d_name) &&
             strcmp("..",dir_entry->d_name)){
                dir_recursivo(entrada,
                        profundidade + 1);
            }
          }else{
            printf("(DIR='%s',P=%d) '%s' (%s)\n",
           nome_dir, profundidade,
                                dir_entry->d_name,
           devolve_tipo_entrada(stat_buf.st_mode));
          }
      }
    }while(finito==0);

    if( closedir(dir_d) == -1 ){
      fprintf(stderr,"erro: impossível fechar DIR"
          "'%s' - %s\n",nome_dir, strerror(errno));
      return -1;
    }
    return 0;
}

int main(int argc, char *argv[]){
    if( argc != 2 ){
      fprintf(stderr,"ERRO\n");
      fprintf(stderr,"%s <diretório raiz>\n",
          argv[0]);
      exit(EXIT_FAILURE);
    }
    int profundidade = 0;
    char *nome_diretorio = argv[1];
    dir_recursivo(nome_diretorio, profundidade);
    return 0;
}
```

Listagem 9: programa dir_recursivo

Reentrância

O facto do código apresentado neste artigo usar a função não reentrante readdir, leva a que não seja seguro usar o código em ambiente multithreaded. Contudo, no caso da função readdir é relativamente fácil substituir o seu uso pela versão reentrante readdir_r. Esta abordagem é deixada como exercício para os leitores, sugerindo-se a consulta do manual eletrónico (man readdir_r).

Nota final

Este artigo abordou a travessia com busca em profundidade de uma árvore de diretórios, com o intuito de visitar todos os recursos (ficheiros, subdiretórios) dessa mesma árvore. Embora o enfoque tenha sido em diretórios, uma metodologia similar pode ser empregue, com as devidas

adaptações, com outros tipos de árvores, por exemplo, árvores binárias.

> ***O acesso ao conteúdo é feito através de chamadas sucessivas à função readdir que devolve uma entrada por chamada (...)***

Bibliografia

- Chris Anley, J. H. (2007). *The Shellcoder's Handbook: Discovering and Exploiting Security Holes* (2nd ed.). Wiley Publishing, Inc.
- Knuth, D. (1968). *The Art of Computer Programming, volume 1: Fundamental Algorithms.* Addison-Wesley.
- Wikipedia. (2015). *Busca em Profundidade.* Obtido em 06 de 11 de 2015, de Wikipedia: https://pt.wikipedia.org/wiki/Busca_em_profundidade

AUTOR

Escrito Por Patrício Domingues

é professor do Departamento de Engª Informática na Escola Superior de Tecnologia e Gestão (ESTG) do Instituto Politécnico de Leiria (IPLeiria). Tem lecionado, entre outras, as disciplinas de Programação Avançada e Sistemas Operativos da Licenciatura em Engenharia Informática.

AUTOR

Escrito Por Vítor Carreira

leciona as disciplinas de Programação Avançada, Sistemas Operativos e Desenvolvimento de Aplicações Distribuídas ao curso de Licenciatura em Engenharia Informática da Escola Superior de Tecnologia e Gestão do Politécnico de Leiria.

AUTOR

Escrito Por Carlos Grilo

é coordenador do Mestrado em Engenharia Informática – Computação Móvel da Escola Superior de Tecnologia e Gestão do Politécnico de Leiria. Leciona as disciplinas de Programação Avançada e Desenvolvimento de Aplicações Empresariais ao Curso de Licenciatura em Engenharia Informática.

A PROGRAMAR

Validação de formulários em JavaScript

Aplicações Mobile com o Xamarin Studio

Plotagem de dados “em tempo real” com Python usando matPlotLib

Xamarin 4 tem tudo que precisa para criar óptimas aplicações móveis!

Validação de formulários em JavaScript

Âmbito

Atualmente, a validação de formulários em JavaScript é muito utilizada para garantir que os dados solicitados nos campos que o utilizador preenche em qualquer site sejam fiéis ao que o programador deseja. A validação de dados por meio de programação JavaScript é feita no browser (sem necessidade de ir até o servidor para transmitir dados) e garante a consistência dos dados quando são enviados para a base de dados.

A validação no cliente é necessária em várias situações: quando, por exemplo, o utilizador não preencheu corretamente o seu e-mail e colocou 2 invés de @. Se esta validação for no cliente, o utilizador recebe a mensagem de e-mail inválido antes do formulário ser enviado. A validação no cliente melhora a interação do utilizador com o site, mas é claro, que as validações no servidor também devem ser feitas.

Introdução

Neste artigo apresento um script completo que exemplifica a validação de um formulário em JavaScript.

Primeiro é necessário criar o formulário que contém os campos a serem validados.

Código HTML:

Segue abaixo o código HTML para criar o formulário.

```
<form name="frmTeste" method="post" action="#"
onsubmit="return validaForm(this);">
<p>Validação de formulário</p>
<p>
    Nome:<input type="text" name="nome"
                                    id="nome" />
</p>
<p>Email:
    <label><input type="text" name="email"
                            id="email" /></label>

</p>
<p>Sexo:
    <label>
        <input type="radio" name="sexo" value="M"
                           id="masc" /> Masculino
    </label>
    <label>
        <input type="radio" name="sexo" value="M"
                          id="fem" /> Feminino
    </label>
</p>
<p>Profissão
<label>
    <select name="cargo">
        <option value = "">Selecione o cargo
                                        </option>
        <option value = "programador">Programador
                                        </option>
        <option value = "designer">Designer
                                        </option>
        <option value = "tester">Tester</option>
        <option value = "todas">Todas</option>
    </select>
</label>
</p>
<label>
    <input type="submit" name="sbt"
                                    value="Enviar" />
</label>
<br />
</p>
</form>
```

O código acima deve ser colocado dentro do corpo do documento <body> </body>. Na tag form o evento onSubmit chama uma função denominada validaForm que é responsável pela validação dos dados do formulário. Se os dados estiverem corretos a função retorna true e o formulário é enviado. Caso contrário é mostrada uma mensagem que determina o local do erro.

Código JavaScript

Segue abaixo o código de validação em JavaScript. Este código deve ser colocado dentro do cabeçalho do documento <head> </head>.

```
<script type="text/javascript">
<!--
function validaform(frm) {
/*
o parametro frm desta função significa: this.form,
pois a chamada da função - validaForm(this) foi
definida na tag form.
*/

    //Verifica se o campo nome foi preenchido e
      //contém no mínimo
    //três caracteres.
    if(frm.nome.value == """ || frm.nome.value ==
      null || frm.nome.value.lenght < 3) {

        //É mostrado um alerta, caso o campo esteja
                                        //vazio.
        alert("Por favor, indique o seu nome.");

        //Foi definido um focus no campo.
        frm.nome.focus();

    //o form não é enviado.
    return false;
    }

    //o campo e-mail precisa de conter: "@", "." e
                            //não pode estar vazio

    if(frm.email.value.indexOf("@") == -1 ||
     frm.email.valueOf.indexOf(".") == -1 ||
     frm.email.value == "" ||
```

```
    frm.email.value == null) {
       alert("Por favor, indique um e-mail
                                        válido.");
       frm.email.focus();
    return false;

/*
O utilizador necessita de selecionar um dos dois
radio buttons: Masculino ou Feminino.
*/
escolhaSexo = -1; //valor negativo default (padrão)
//que significa que nada foi escolhido ainda.

/*
No bloco de código abaixo foi criado um ciclo entre
os radios button
com o mesmo nome (sexo)
*/
for (x = frm.sexo.lenght -1; x > -1; x--) {
/*
x = frm.sexo.lenght -1 é a mesma coisa que: x = 2-
1, que resulta em 1.
x > -1 significa que o valor de x não pode ser
igual a -1 e
sim maior, porque -1 significa que nada foi
escolhido.
x-- significa que há um decremento no valor x, é
algo como:
x = 1, x= 0 e pára pois x não pode ser -1.
*/

    if(frm.sexo[x].checked) { //checked quer dizer
                              //selecionado,
    //então verifica se o primeiro (0)  ou o
    //segundo (1) radio button
    //foi selecionado (checked).
        escolhaSexo = x; //atribui à variável
                         //escolhjaSexo o valor X.
    }
}
/*
se nenhuma das opções (masculino ou feminino) forem
escolhidas, mostra um alerta e cancela o envio.
*/
if (escolhaSexo == -1) {
    alert("qual o seu sexo?");
    frm.sexo[0].focus();
return false;
}

/* valida a profissão:
O utilizador tem de escolher uma entre as três
opções (Programador, Designer e Tester).
*/
if(frm.prof.value == "" || from.prof.value ==
                                        "Todas") {
    alert("De momento, precisamos de especialistas
                        nas três áreas indicadas");
    frm.prof.focus();
return false;
}

// Valida a textArea, que é como validar um campo
                                //de texto simples.
if(frm.sobre.value == "" || frm.sobre.value ==
                                                null)
{
    alert("Por favor, conte-nos um pouco sobre
si.");
    frm.sobre.focus();
return false;
}
</script>
```

Conclusão

Esta foi uma breve demonstração de como podemos implementar validações para os nossos formulários HTML usando apenas JavaScript.

> **A validação no cliente é necessária em várias situações: quando, por exemplo, o utilizador não preencheu corretamente o seu e-mail e colocou 2 invés de @ . (...)**

É importante realçar que a validação de formulários que são submetidos ao servidor web não deve depender apenas de JavaScript no lado cliente, pois não se pode assegurar que o recurso esteja ativado e a funcionar corretamente em todos os browsers. Implementada ou não a validação em JavaScript, deve sempre haver a validação dos dados recebidos **no servidor** da aplicação. O uso de validação via JavaScript cliente serve essencialmente para facilitar a **pré-validação no lado cliente**, possibilitando ao utilizador ter os seus dados verificados ainda antes de submeter o formulário.

AUTOR

Escrito por Tânia Valente

Natural de Coimbra, licenciou-se em Engenharia Informática pelo Instituto Superior de Engenharia de Coimbra e, actualmente, frequenta o mestrado em Human Computer Interaction. É entusiasta na área de Desenvolvimento Web, no que concerne às Tecnologias Web, Design de Interface (UI) e User Experience (UX). Curiosa e motivada por novos desafios, acredita que a criatividade pode transformar a maneira como as pessoas pensam, sentem e agem.

Aplicações Mobile com o Xamarin Studio

Introdução

Uma das linguagens de programação que tem tido um aumento susceptível a nível de popularidade, é o C#. Cada vez mais, esta tem sido a escolha feita pelos programadores para por as suas ideias em prática, nos mais variados tipos de software. Chegou a altura de esta ter uma palavra a dizer no que toca a aplicações Android e IOS, visto que só era aplicada para apps Windows Phone. Para ajudar a festa, temos o Xamarin Studio.

Descrição

O Xamarin Studio é um IDE que traz consigo um vasto conjunto de features, o que irá ajudar ao desenvolvimento de aplicações visualmente atractivos e com relativa facilidade, utilizando todo o poder do C#. Neste artigo, vou mostrar como podem criar uma aplicação para dispositivos Android, uma app de introdução muito simples. Convém verificar durante a instalação do IDE se o Android SDK é instalado (é instalado normalmente junto com o Xamarin Studio), pois sem ele não vamos poder criar qualquer tipo de projecto.

Figura 1: Welcome Screen do Xamarin Studio

O processo de criação de um projecto de aplicações Android no Xamarin, não foge muito ao que é "normal" em todos os IDEs. A imagem abaixo mostra os passos por ordem numérica que são necessários para a criação do mesmo.

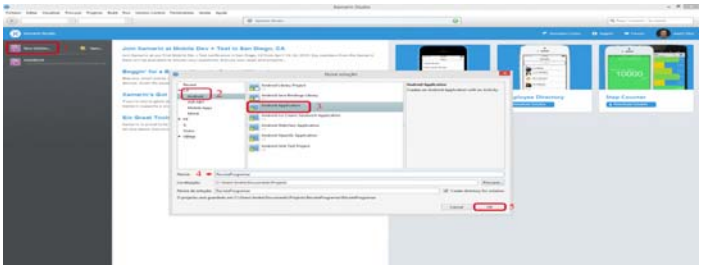


Figura 2: Passos ordenados de 1 a 5 para a criação do projecto Android

Depois deste processo, é criado o projecto na qual vamos trabalhar. Digamos que os utilizadores de Android Studio, não deverão ter dificuldades em perceber a estrutura do projecto em si, pois são muito semelhantes. No lado esquerdo da janela do IDE, está a Solução ou o Projecto propriamente dito (ver figura 3).

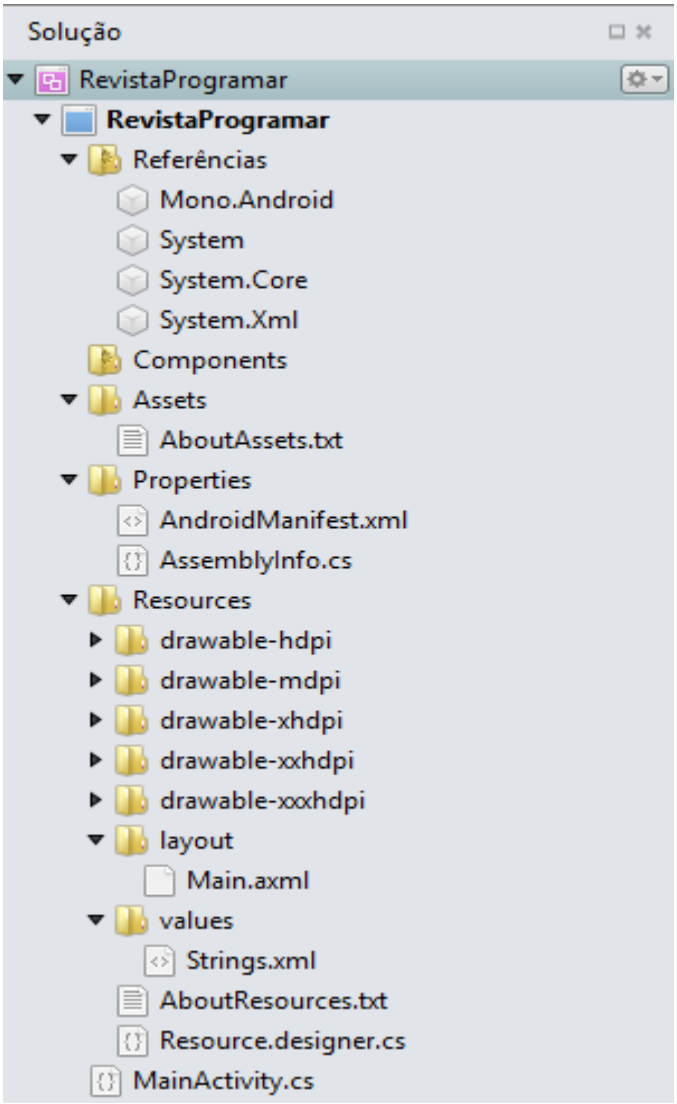


Figura 3: Estrutura do projecto

Na Solução, podem ver que os ficheiros apresentados são familiares a programadores Android, por isso, neste artigo não vou alongar muito mais a explicação dos mesmos.

Vamos passar a aplicação propriamente dita. Para podermos ver o layout da aplicação assim que o projecto é criado, basta fazer abrir a pasta layout e fazer double-click no Main.xml. Assim poderão ver como está a vossa aplicação em termos de apresentação. Na mesma janela, há uma

opção para editar o layout por drag and drop, Content (ver figura 5), ou então por código puro, Source (ver figura 4).

Figura 4: Source

Figura 5: Content

Em seguida, na estrutura do projecto, vamos abrir o ficheiro onde está o principal código da aplicação, o MainActivity.cs. Por defeito, o Xamarin já cria código de exemplo, para que a aplicação corra de imediato no emulador ou dispositivo físico. Explicando rapidamente o que esse código faz, é uma função que contabiliza o número de cliques que são dados num botão, aprestando os dados dentro do mesmo. Poderão ver isso na figura abaixo (ver figura 6).

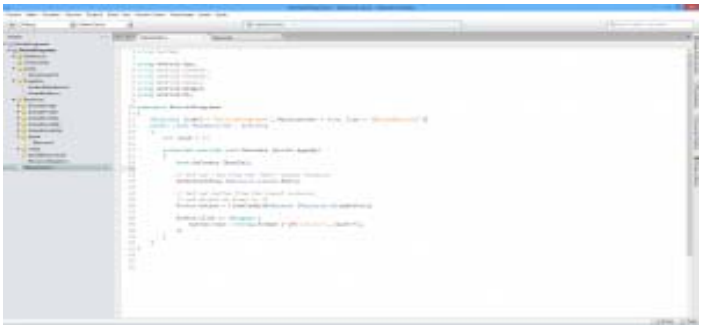


Figura 6: Source Code

Neste artigo, não vou aprofundar nenhum código específico, ou desenvolver algo, sendo que, o que falta aqui é ver a aplicação a funcionar. Para tal, vamos ter de ir à barra de menu do Xamarin, clicar em *Ferramentas* e no sub-menu escolher a opção *Android Devices...*. Abrirá uma pequena janela com uma lista dos emuladores que já estão predefinidos no IDE, mas vamos criar o nosso próprio. O processo é exactamente o mesmo que no Android Studio, ou no SDK. Depois de o emulador estar criado, vamos executar a nossa aplicação, simplesmente carregando no botão *Start*, no canto superior esquerdo do nosso IDE. Assim, desta maneira vamos ver a app em acção (ver figuras 7 e 8).

Figura 7: Aplicação iniciada

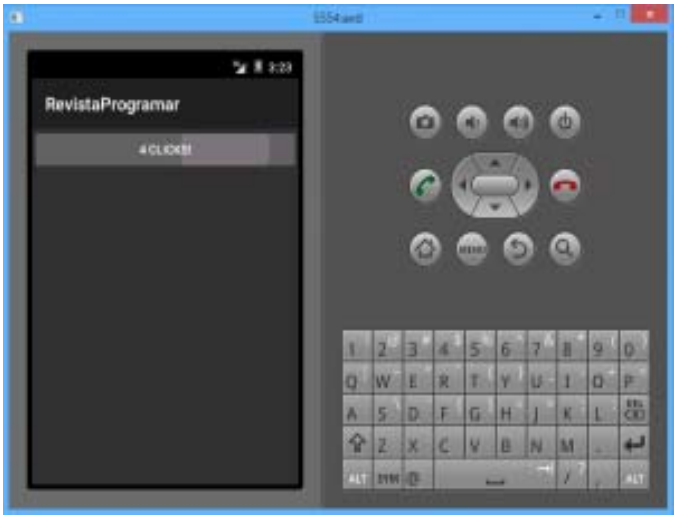
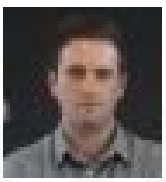


Figura 8: Aplicação em execução

Conclusão

Com o Xamarin, é muito fácil começar um projecto para qualquer aplicação Android ou IOS. Agora é dar asas à imaginação e começar a programar.

AUTOR



André Silva

Entusiasta das tecnologias desde pequeno, decidiu seguir a vida de programador. Formado em Gestão e Programação de Sistemas Informáticos e Programação Android. Neste momento é Co-Fundador da Digital Soul Games. digitalsoulgames.org

Plotagem de dados "em tempo real" com Python usando matPlotLib

Introdução

Uma das tarefas mais comuns quando se trabalha com aquisição de dados, é a representação dos mesmos em gráficos, de forma a serem mais facilmente interpretáveis pelo utilizador. Na verdade seja para um uso de hobbie, seja para uma utilização mais científica ou profissional, na maior parte dos casos, acabamos quase sempre por preferir a leitura gráfica dos dados, o que nos permite rapidamente uma noção mais ampla do contexto visível.

Como é do conhecimento do leitor, hoje em dia existe uma grande quantidade de dispositivos que transmitem dados via USB, usando conversão USB – Série, para a transmissão dos dados adquiridos, referentes a um ou mais parâmetros num contexto ou ambiente. No entanto o que a maioria dos dispositivos tem em comum, é a forma como transmite os dados para o computador, normalmente usando um cabo USB ou série e transmitindo os dados em formato de texto simples. Ao longo deste artigo irá ser apresentado o código para receber dados de um conjunto de quatro sensores independentes, ligados a um mesmo dispositivo que os transmite via RS232 e posterior apresentação em modo gráfico, usando a linguagem Python e a biblioteca matplotlib, para nos desenhar o gráfico em tempo "quase real". (Para os mais distraídos a transmissão via RS-232 é um padrão de protocolo para troca serial de dados binários entre um DTE - *Data Terminal equipment* e um DCE *Data Communication equipment*. Normalmente usado nas portas séries dos computadores.)

Neste artigo não iremos focar o software que corre no dispositivo, apenas a aquisição de dados e a plotagem dos mesmos, sendo que os diagramas do circuito, bem como o software que corre no micro-controlador, saem claramente do âmbito do que é pretendido neste artigo.

A biblioteca matplotlib, fornece a capacidade de representar dados em gráficos cartesianos, entre outros sendo quase uma alternativa livre, a outras peças de software, mais complexas e proprietárias. Além da matplotlib iremos usar a biblioteca *numpy*, para a parte de cálculos que iremos realizar antes da plotagem dos dados.

O código

Ao longo dos próximos parágrafos, iremos apresentar o código, que será utilizado no final para realizar a plotagem, *isto* é, a impressão de desenhos em larga escala,dos dados. Em todo o artigo o código escrito é python versão 2.7, pelo que se recomenda ao leitor que siga esta mesma versão.

Posto isto, vamos ao nosso programa! Para isso começamos por importar a biblioteca que nos permitirá lidar com as comunicações, as duas bibliotecas que nos irão permitir fazer o plot dos dados, primeiro a que nos irá permitir desenhar os gráficos em "tempo real" e por fim a que nos permite armazenar os dados em memória, pelo tempo necessário à sua utilização no âmbito desta aplicação:

```
import serial # carrega a biblioteca Serial para
              # comunicações série
import numpy  # carrega a biblioteca numpy
import matplotlib.pyplot as plt #carrega a
                                #biblioteca pyplot
import matplotlib.gridspec as gridspec #carrega a
                               # biblioteca gridspec
from drawnow import * #carrega a biblioteca
                     # drawnow
from array import array #carrega a biblioteca
                        # array
```

A biblioteca gridspec, será utilizada para disponibilizar múltiplos gráficos numa mesma "janela", cada gráfico com unidades diferentes e escalas diferentes. Neste caso teremos quatro grupos de gráficos, disponibilizados em "tempo real", pelo que a biblioteca gridspec nos vai permitir dispô-los na janela de forma conveniente e controlar os gráficos de forma independente, como iremos ver mais adiante neste artigo.

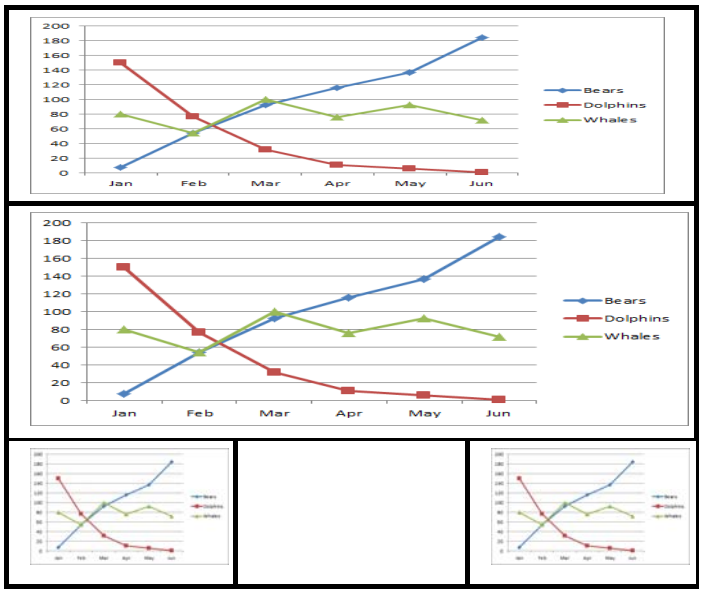
Feito o carregamento das bibliotecas, precisamos de criar os objectos onde vamos armazenar os valores, antes mesmo de iniciarmos a comunicação com o dispositivo. Neste exemplo usaremos seis vectores de variáveis do tipo float. Para comunicar via série, iremos precisar de um objecto do tipo serial, que recebe dois parâmetros, neste caso o nome da porta série e a velocidade de comunicação, também conhecida por *baud rate.* Por fim vamos usar um contador, pois não vamos querer todos os dados armazenados em memória, apenas as últimas posições e iniciamos a matplotlib em modo interactivo utilizando o método *ion().*

```
#instanciação dos objectos
tempF = array('f')
tempF2 = array('f')
humF1 = array('f')
humF2 = array('f')
lum = array('f')
moist = array('f')
#comunicacao
arduinoData = serial.Serial('com3', 115200) #Cria
# um objecto do tipo serial chamado
# arduinoData
plt.ion()
cnt=0
```

Finda esta parte, vamos agora criar a função que irá ser chamada a cada nova leitura de dados, para desenhar e actualizar o gráfico com os novos dados obtidos.

Nesta função, chamemos-lhe "makeFig", iremos desenhar uma grelha de 3 por 3 recorrendo à biblioteca gridspec, a fim de suportar a disposição dos 4 gráficos apresenta-

dos, deixando espaço entre eles. A grelha será algo semelhante à figura abaixo:



Criada a grelha, cada gráfico passa a ser uma "imagem", separada, com as suas propriedades independentes, como o caso da posição, legendas, escalas, etiquetas, símbolos e cores das linhas. De igual modo cada um dos restantes quatro gráficos, terá essa mesma definição, como pode ser observado no código da função. Para simplificar, apenas está comentado o código da primeira "figura", acreditando que para o leitor, será simples entender o restante código, uma vez que as instruções são as mesmas, mudando apenas alguns parâmetros, específicos para cada figura.

```
def makeFig():
    gs = gridspec.GridSpec(3, 3) #gridspec 3x3
    #Plot 1
    plt.subplot(gs[0, :])#posicao do subplot
    plt.ylim([-30,50])#valor min e max de y
    plt.title('Temperatura em Graus C')#titulo
    plt.grid(True)
    plt.ylabel('Temp-1 c')#etiquetas do eixo y
    plt.plot(tempF, 'ro-', label='temperatura em
                    graus')    #plot de temperature
    plt.legend(loc='upper left')#plot da legenda
    plt2=plt.twinx()#cria um Segundo eixo y
    plt.ylim(-30,50)#define os limites do

#Segundo eixo y
    plt2.plot(tempF2, 'b^-', label='Temp-2 c')
                        # desenha os val. De tempF2
    plt2.set_ylabel('Temp-2 c') #Etiqueta do
                                # Segundo eixo
    plt2.ticklabel_format(useOffset=False)# impede
                               # a escala do eixo X
    plt2.legend(loc='upper right')
    #Plot 2
    plt.subplot(gs[1, :])
    plt.ylim([0,100])
    plt.title('Humidade do Ar em Percentagem')
    plt.grid(True)
    plt.ylabel('Humidade -1 %')
    plt.plot(humF1, 'ro-', label='Humidade')
    plt.legend(loc='upper left')
    plt2=plt.twinx()
    plt.ylim(0,100)
    plt2.plot(humF2, 'b^-', label='Humidade-2 %')
    plt2.set_ylabel('Humidade-2 %')
    plt2.ticklabel_format(useOffset=False)
    plt2.legend(loc='upper right')
    #Plot 3
    plt.subplot(gs[-1,0])
    plt.ylim([0,100])
    plt.title('Humidade do solo')
    plt.grid(True)
    plt.ylabel('Humidade %')
    plt.plot(moist, 'ro-', label='Humidade')
    plt.legend(loc='upper left')
    #Plot 4
    plt.subplot(gs[-1,-1])
    plt.ylim([0,2000])
    plt.title('Luminosidade')
    plt.grid(True)
    plt.ylabel('Luminosidade (lux)')
    plt.plot(lum, 'ro-', label='Luminosidade')
    plt.legend(loc='upper left')
```

Chegados a esta parte do código, falta-nos escrever o programa principal que será executado. Um dos problemas mais comuns com aquisição de dados via série, são os atrasos e as falhas de transmissão de dados. Para evitar algumas dessas situações podemos indicar ao computador que a menos que existam dados para serem recebidos, não executa o restante código. Neste exemplo concreto usamos um ciclo while para o fazer, verificando se existem dados no buffer. Se existirem, o restante código é executado, caso contrário repete o ciclo, aguardado a disponibilidade dos mesmos.

```
while (arduinoData.inWaiting()==0): #aguarda por
                                    # dados
        pass #não executa nada
```

Existindo dados no buffer, procedemos à leitura dos mesmos recorrendo ao método readline(), como veremos no código exemplo. Neste caso convém alertar para o facto de os dados virem em formato de texto (string) e ser necessária a sua conversão a float, para ser feito o plot. Adicionalmente os dados vêm numa única linha, separados por vírgula (,), noutros casos podem utilizar qualquer outro caracter de separação, dependendo do sistema utilizado. A primeira tarefa a fazer, após a recepção da informação, será a separação da string, nas múltiplas variáveis e posterior armazenamento dos dados em memória ram, antes de fazermos a sua conversão para float e armazenamento no vector de valores do tipo float, que será usado para gerar o gráfico.

Uma vez que as dimensões de um ecrã, são limitadas, vamos optar por manter apenas visíveis os últimos cinquenta pontos de cada gráfico, seguindo uma lógica FIFO (first in, first out), onde os valores que "entraram" primeiro (mais antigos), são eliminados e os novos são acrescentados uma posição à frente da última posição utilizada recorrendo ao método pop(), que recebe como argumento a posição que queremos eliminar o valor nela contido.

Como o desenho será em "tempo real", teremos de colocar todas estas tarefas num ciclo infinito, por forma a manter a sua execução ininterrupta, neste caso, novamente um ciclo while. Poderíamos ter usado qualquer outro tipo de

ciclo, mas por uma questão de comodidade, usou-se um while, deixando-se ao critério do leitor, usar qualquer outro ciclo da sua preferência.

Abaixo apresentamos a listagem completa do código até agora construído.

```
#graphics in python
import serial # carrega a biblioteca Serial para
              # comunicações série
import numpy  # carrega a biblioteca numpy
import matplotlib.pyplot as plt #carrega a
                            # biblioteca matplotlib
import matplotlib.gridspec as gridspec #carrega a
                                # biblioteca gridspec
from drawnow import * #carrega a biblioteca drawnow
from array import array #carrega a biblioteca array

tempF = array('f')
tempF2 = array('f')
humF1 = array('f')
humF2 = array('f')
lum = array('f')
moist = array('f')

arduinoData = serial.Serial('com3', 115200)
plt.ion()

#function makeFig()
def makeFig():
    gs = gridspec.GridSpec(3, 3)
    #Plot 1
    plt.subplot(gs[0, :])
    plt.ylim([-30,50])
    plt.title('Temperatura em Graus C')
    plt.grid(True)
    plt.ylabel('Temp-1 c')
    plt.plot(tempF, 'ro-', label='temperatura em
                                            graus')
    plt.legend(loc='upper left')
    plt2=plt.twinx()
    plt.ylim(-30,50)
    plt2.plot(tempF2, 'b^-', label='Temp-2 c')
    plt2.set_ylabel('Temp-2 c')
    plt2.ticklabel_format(useOffset=False)
    plt2.legend(loc='upper right')
    #Plot 2
    plt.subplot(gs[1, :])
    plt.ylim([0,100])
    plt.title('Humidade do Ar em Percentagem')
    plt.grid(True)
    plt.ylabel('Himidade-1 %')
    plt.plot(humF1, 'ro-', label='Humidade')
    plt.legend(loc='upper left')
    plt2=plt.twinx()
    plt.ylim(0,100)
    plt2.plot(humF2, 'b^-', label='Humidade-2 %')
    plt2.set_ylabel('Humidade-2 %')
    plt2.ticklabel_format(useOffset=False)
    plt2.legend(loc='upper right')
    #Plot 3
    plt.subplot(gs[-1,0])
    plt.ylim([0,100])
    plt.title('Humidade do solo')
    plt.grid(True)
    plt.ylabel('Himidade %')
    plt.plot(moist, 'ro-', label='Humidade')
    plt.legend(loc='upper left')
    #Plot 4
    plt.subplot(gs[-1,-1])
    plt.ylim([0,2000])
    plt.title('Luminosidade')
    plt.grid(True)
    plt.ylabel('Luminosidade (lux)')
    plt.plot(lum, 'ro-', label='Luminosidade')
    plt.legend(loc='upper left')

#mainProgram
while True:
    while (arduinoData.inWaiting()==0):
        pass
    arduinoString = arduinoData.readline()
    #"A tarefa mais urgente da vida, é: O que
    # estás a fazer pelos outros?" (Martin
    # Luther King Jr.)
    splitedArray = [float(s) for s in
                        arduinoString.split(',')]
    temp = splitedArray[0]
    hum1 = splitedArray[1]
    temp2 = splitedArray[2]
    hum2 = splitedArray[3]
    moisture = splitedArray[4]
    lumen = splitedArray[5]
    tempF.append(temp)
    tempF2.append(temp2)
    humF1.append(hum1)
    humF2.append(hum2)
    moist.append(moisture)
    lum.append(lumen)
    drawnow(makeFig)
    plt.pause(.000005)
    cnt=cnt+1
    if(cnt>50):
        tempF.pop(0)
        tempF2.pop(0)
        humF1.pop(0)
        humF2.pop(0)
        moist.pop(0)
        lum.pop(0)

#"The best way to learn is to
# teach" (Openhimmer)
```

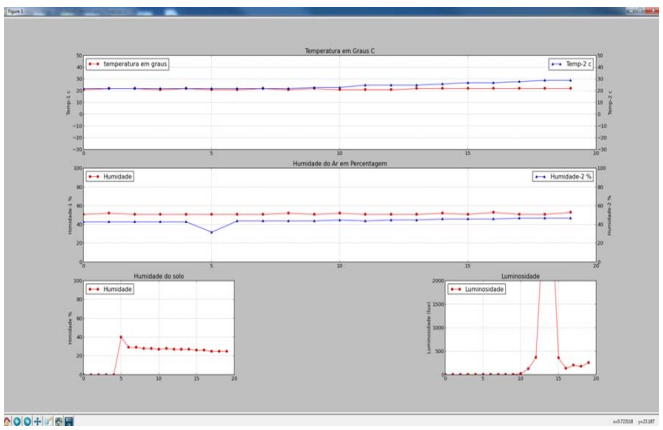


Imagem do programa em execução

Conclusão

Ao longo deste artigo apresentamos a matplotlib e as restantes bibliotecas de python que utilizamos para gerar gráficos em tempo real. Exploramos a criação de múltiplos gráficos com escalas diferentes numa mesma figura, e por fim a recepção dos dados, provenientes de um equipamento que comunique com o computador por porta série, ou qualquer emulação dessa mesma porta e a sua representação gráfica, em cada um dos gráficos do layout, de forma independente. Como pudemos ver, é possível criar os mais varia-

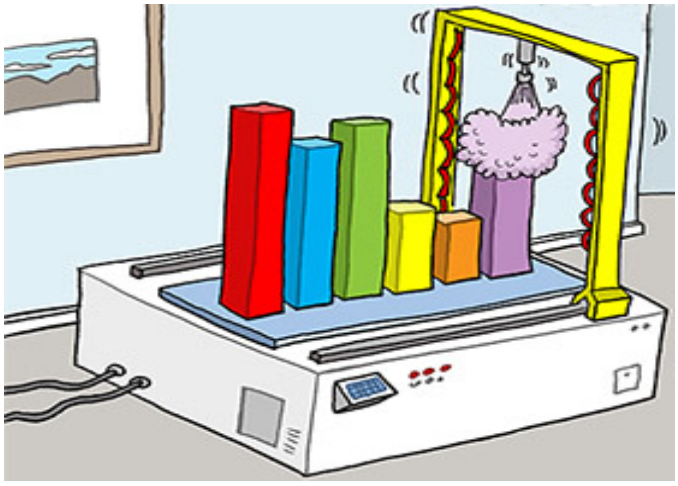
dos layouts para a apresentação de gráficos usando a linguagem de programação python e a biblioteca matplotlib. É de facto simples comunicar via porta série usando a biblioteca *serial* e desenhar objectos gráficos no ecrã recorrendo ao conjunto de bibliotecas *drawnow*.

> ***Uma das tarefas mais comuns quando se trabalha com aquisição de dados, é a representação dos mesmos em gráficos, de forma a serem mais facilmente interpretáveis pelo utilizador (...)***

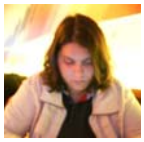
Deixamos as referências que utilizamos neste artigo e esperamos que o leitor se sinta entusiasmado a experimentar a linguagem python e as bibliotecas referidas no exemplo.

Referências

- Documentação oficial da biblioteca matplotlib: http://matplotlib.org/
- Documentação oficial da biblioteca numpy: http://www.numpy.org/
- Documentação oficial da biblioteca pySerial: https://pythonhosted.org/pyserial/
- Python- Algoritmia e Programação Web (José Braga de Vasconcelos) FCA Editora
- Scientific Computing in Python – NumPy, SciPy, Matplotlib (Dr. Axel KohlMeyer) http://www.ictp-saifr.org/wp-content/uploads/2014/09/numpy-scipy-matplotlib.pdf
- Instrument Control (iC) – An Open-Source Software to Automate Test Equipment (NIST) Volume 117 (2012) http://dx.doi.org/10.6028/jres.117.010 Journal of Research of the National Institute of Standards and Technology
- Real World Instrumentation with Python (J. M. Hughes) O'Reilly



AUTOR



Escrito por Rita Peres

Natural de Castelo Branco, licenciou-se em Engenharia Informática pela Universidade da Beira Interior. Membro do P@P desde Janeiro de 2010.

AUTOR

Escrito por António C. Santos

Com uma enorme paixão por tecnologia, autodidacta desde tenra idade, cresceu com o ZX Spectrum. Tem vasta experiência em implementação e integração de sistemas ERP, CRM, ERM, BI e desenvolvimento de software por medida nas mais diversas linguagens. Diplomado do Curso de Especialização Tecnológica em Tecnologias e Programação de Sistemas de Informação pela ESTG-IPVC. Membro da Comunidade Portugal-a-Programar desde Agosto de 2007, é também membro da Sahana Software Foundation, onde é Programador Voluntário. Neste momento é aluno no Instituto Politécnico de Viana do Castelo, na Escola Superior de Tecnologia e Gestão no curso de Licenciatura em Engenharia Informática e Xamarin Student Partner nesta mesma escola Twitter:@apocsantos

Xamarin 4 tem tudo que precisa para criar óptimas aplicações móveis!

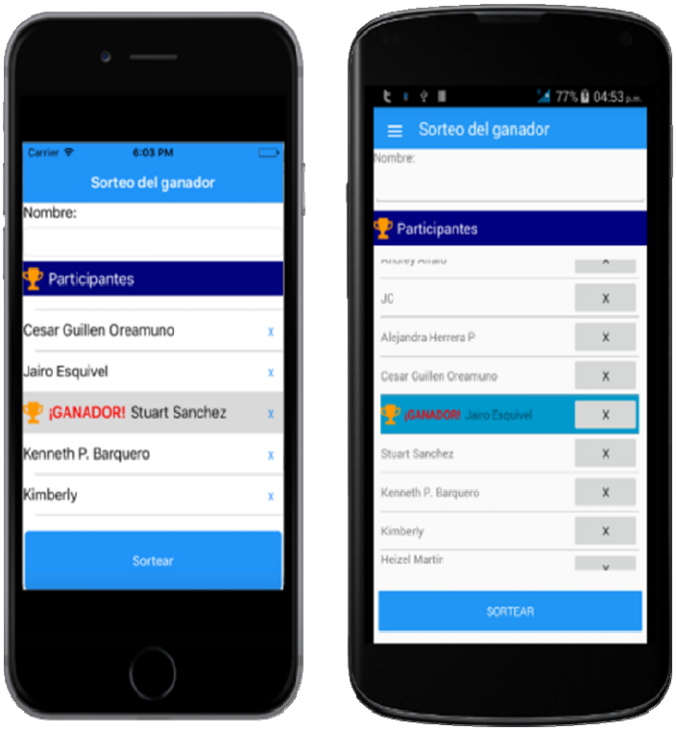
Vamos dar uma olhada nalgumas das funcionalidades e melhorias incluídas no Xamarin 4.0 para o desenvolvimento de aplicações multi-plataformas, incluindo Android, iOS e UWP (Universal Windows Apps).

Apresentando Xamarin Forms 2.0

Desta vez começamos por um aplicativo simples chamado " Lucky Winner" para selecionar aleatoriamente uma pessoa a partir de uma lista de participantes usando a classe System.Random comumente usada para gerar números aleatórios tal como em qualquer outro programa para .net.

Aqui estão alguns screenshots do aplicativo em execução em diferentes plataformas:

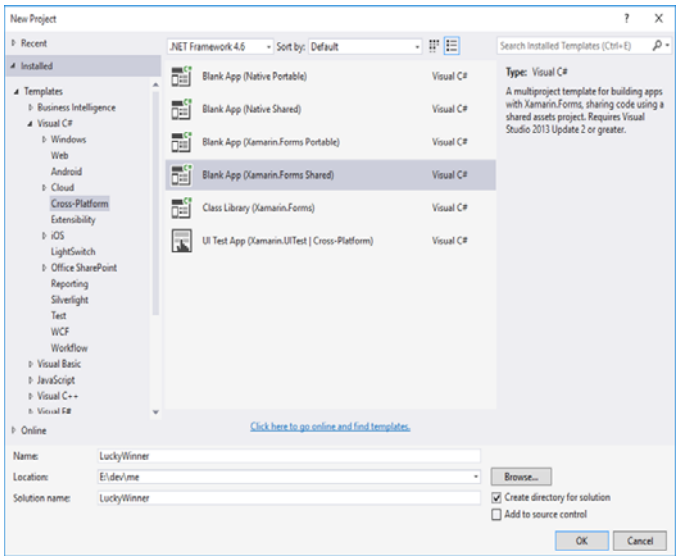
O código fonte completo para este aplicativo pode ser descarregado do meu repositório GitHub aqui.



Desenvolvendo um aplicativo Xamarin.Forms a partir do zero:

Ao instalar a platforma Xamarin e obter uma licença válida ou trial seremos capaz de criar uma PCL (Biblioteca de Classes Portável) que contém a maior parte do código para que possa ser compartilhado e compilado de forma nativa para Android, iOS e UWP.

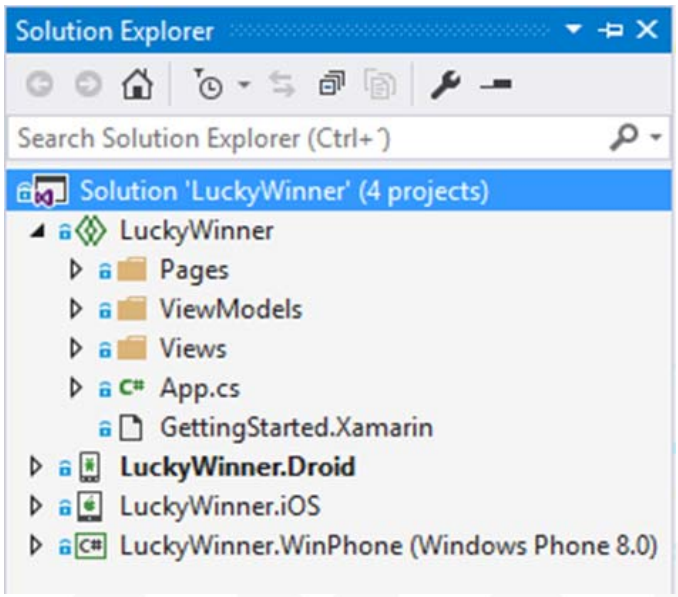
Então, vamos escolher o modelo Xamarin.Forms da lista disponível dentro do Visual Studio:



Ao fazê-lo iremos receber:

- projeto Xamarin.Forms A para a partilha de código XAML / C # entre plataformas
- projeto de referência Android Xamarin.Forms
- projeto iOS
- projeto UWP

Se o leitor estiver familiarizado com MVVM, XAML e C # será bastante familiar para incluir facilmente Pages, ViewModels, Views no projeto como na imagem seguinte.



Construção de interfaces com MVVM, de uma forma multi-plataforma:

Vamos dar vista de olhos no código XAML usado para definir a lista de participantes:

```
<ListView x:Name="PlayersSelector"
          ItemsSource="{Binding Path=Players}"
          SeparatorColor="Gray"
          SelectedItem="{Binding Path=Winner,
                                 Mode=TwoWay}">
      <ListView.ItemTemplate>
          <DataTemplate>
            <ViewCell>
              <ViewCell.ContextActions>
                  <ToolbarItem Command="{Binding
                    DeleteCommand}" Text="Borrar" />
              </ViewCell.ContextActions>
              <Grid>
                  <StackLayout
                            Orientation="Horizontal"
                           HorizontalOptions="Start">
                    <Image IsVisible="
                              {Binding Path=IsWinner}"
                           Source="Prize.png"
                           HeightRequest="25"
                                  WidthRequest="25" />

                        <Label Text="¡GANADOR!"
                            IsVisible="{Binding
                                        Path=IsWinner}"
                             TextColor="Red"
                             FontAttributes="Bold"
                    VerticalTextAlignment="Center"/>

                        <Label Text="{Binding
                                      Path=PlayerName}"
                     erticalTextAlignment="Center"/>
                  </StackLayout>
                  <Button HorizontalOptions="End"
                       Command="{Binding
                                 Path=DeleteCommand}"
                       Text="x" />
              </Grid>
              </ViewCell>
        </DataTemplate>
      </ListView.ItemTemplate>
</ListView>
```

O *Binding* atua como cola entre a View e os ViewModels para que permitir ter uma clara separação de tarefas entre eles. Outras características comuns suportados pelo Xamarin.Forms com MVVM incluem *Two-way binding*, *Command Pattern* and *Value Converters*.

Obter um "Lucky winner" com C # tal como se faria em qualquer outro programa.

Lembra-se como se faz para gerar um número aleatório em C # ? Mas não temos a certeza se sabemos fazê-lo para uma aplicação mobile? Não existe preocupação! Podemos usar os recursos principais do sistema padrão em C # (até à versão 6.0) além de outros recursos interessantes, incluindo expressões lambda, delegates, actions, Linq e operações assíncronas.

```
private void Play()
{
    ...
    var random = new Random
                      (DateTime.Now.Millisecond);

    var lucky = random.Next(0,
                        ViewModel.Players.Count);
    var selectedPlayer =
     ViewModel.Players.ElementAtOrDefault(lucky);

    if (selectedPlayer != null)
    {
        selectedPlayer.IsWinner = true;
        ViewModel.Winner = selectedPlayer;

        PlayersSelector.ScrollTo(selectedPlayer,
             ScrollToPosition.MakeVisible, true);
    }
}
```

Trazer funcionalidades de desenho para as aplicações Android

Outra grande característica de Xamarin para Android é a capacidade de incorporar facilmente capacidades de desenho nos aplicativos do Android. Isto pode ser feito através da adição da biblioteca Support Design Library disponível em pacote de NuGet.

Recursos adicionais e links

- Introducing Xamarin 4
- Adding Material Design to Xamarin Apps
- Lucky Winner source code

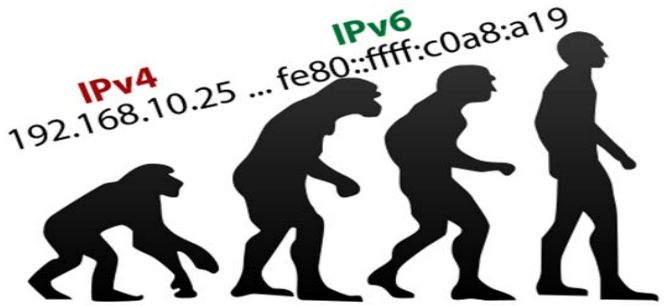
«Traduzido para a Revista PROGRAMAR por António Santos, com permissão e acompanhamento do autor do artigo»

AUTOR

Escrito por Esteban Solano

Xamarin, Web & Full Stack Software Engineer, Founder of the Costa Rica Mobile .Net Developer Group www.meetup.com/es/Xamarin-Costa-Rica-Mobile-NET-Developer-Group/

IPv6 para Programadores



O crescimento da Internet

A Internet como a conhecemos hoje teve as suas origens na ARPANET [1], a primeira rede a implementar o conjunto de protocolos IP (Internet Protocol) [2], do qual o mais conhecido é o TCP (Transmission Control Protocol) [3].

Pretendia-se um protocolo que transmitisse informação, dividida em pacotes de tamanho reduzido, e cuja gestão de rede fosse descentralizada, já que em tempo de Guerra Fria havia a possibilidade de destruição de partes da rede.

Foi utilizada a versão 4 do protocolo IP (IPv4). Na altura estimava-se um número reduzido de máquinas ligadas à rede, mas 40 anos depois a situação é bem diferente:

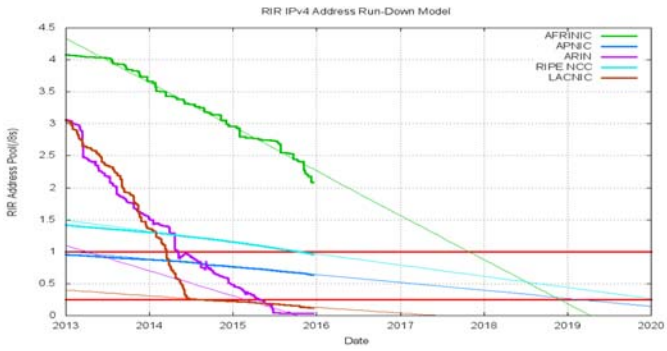
Data	Utilizadores ligados à Internet
1995-12	16 M
2000-12	361 M
2005-12	1018 M
2010-09	1971 M
2015-12 (est.)	3353 M

Fonte: http://www.internetworldstats.com/emarketing.htm

No início dos anos 1990s, e até hoje, dá-se um crescimento continuado do número de pessoas ligadas à Internet, que actualmente se situa perto de 50% da população mundial. Adicionalmente, cada utilizador tem mais que um aparelho (computador, telemóvel, etc.). Mais recentemente (e apesar de o conceito não ser novo), surge a Internet of Things, cuja ideia é ligar todos os aparelhos do dia-a-dia à Internet (televisão, frigorífico, carro, elevador, relógio, roupa, animal de estimação, etc.).

O protocolo IPv4 tem uma grave limitação de endereçamento: só suporta 2^{32} ou 4294967296 endereços (teóricos) mas devido a limitações técnicas, apenas se podem atribuir pouco mais de 3000 milhões, ou seja, **há actualmente menos endereços públicos disponíveis que o número de utilizadores e máquinas/dispositivos**.

A IANA (Internet Assigned Numbers Authority) é a entidade responsável pela atribuição de endereços IPv4 e IPv6, assim como outras numerações (e.g. portas dos protocolos TCP e UDP, etc.). Esta, por sua vez, delega em 5 RIRs (Regional Internet Registries). O RIPE NCC é responsável pela atribuição de endereços na Europa (incluindo toda a Rússia) e Médio Oriente. Esta é a lista de endereços IPv4 actualmente disponíveis:



Nota: Um bloco "/8" corresponde a 2^{24} ou 16777216 endereços

Fonte: http://www.potaroo.net/tools/ipv4/plotend.png (2015-12)

Devido a uma política conservadora de atribuição de endereços, assim como de recuperação de endereços não utilizados, a disponibilidade do RIPE tem-se mantido estável, actualmente com cerca de 15 milhões de endereços IPv4 disponíveis. Já a ARIN, o RIR da América do Norte, esgotou totalmente há alguns meses. Isto significa que os operadores que pretendam ligar-se à Internet nesse continente terão que comprar os blocos de endereços num mercado especial (com custos muito elevados).

IPv6, o sucessor do IPv4

Desde o final dos anos 1980s que este problema tem sido analisado. Várias alternativas para substituição do IPv4 foram propostas:

Versão IP (*)	Nome	Estado
0 a 3	Internet Protocol tests	Experimental / Obsoleta
4	**Internet Protocol Version 4 (IPv4)**	**Ainda em funcionamento**
5	Internet Stream Protocol	Experimental
6	**Internet Protocol Version 6 (IPv6)**	**Em funcionamento / Actual**
7	TP/IX	Experimental / Obsoleta
8	PIP	Experimental / Obsoleta
9	TCP and UDP with Bigger Addresses (TUBA)	Experimental / Obsoleta
10 a 15	Reservada	Reservada

(*) Corresponde ao campo de Versão do cabeçalho dum pacote IP, cuja dimensão é 4 bits.

Após vários anos de debate e análise, as instituições que geriam a Internet optaram pelo IPv6 como melhor alternativa ao IPv4. Devido à lenta adopção do IPv6, foi também criado o protocolo NAT (Network Address Translation) [4] que permite partilhar um único endereço IPv4 público com vários dispositivos que usam endereçamento privado (e.g. 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, entre outros).

O IPv6 trouxe algumas melhorias. Estas são apenas algumas:

- Endereçamento maior:
  - Permite 2^{128} ou 340282366920938463374607432768211456 endereços, o suficiente para atribuir um endereço a cada átomo do universo:
  - Efectivamente apenas 2^{64} ou 18446744073709551616 endereços são atribuídos publicamente, mas mesmo assim mais que suficientes;
  - Os outros 2^{64} endereços (ou mais) são atribuídos por cliente, também suficientes para todas as nossas necessidades em dispositivos IoT;
  - NAT é desnecessário: todos os dispositivos têm endereço IPv6 público.
- Transmissão mais rápida:
  - Não se dá fragmentação de pacotes (i.e., necessidade de dividir os pacotes se excederem a MTU [5] de cada troço da rede por onde passa). A MTU típica de cada pacote em redes Ethernet é de 1500 bytes, embora seja possível usar Jumbo Frames até 9000 bytes. O tamanho máximo de cada pacote é obtido quando se inicia uma ligação (usando Path MTU Discovery [6]), evitando a fragmentação de pacotes e reduzindo assim a latência;
  - A checksum dos pacotes de IPv4 incluíam todos os campos do cabeçalho, o que obrigava a recalcular em cada *hop* (nó da rede). Em IPv6 a checksum aplica-se apenas aos campos do cabeçalho que não se alteram desde a origem até ao destino, evitando a necessidade de recalcular, reduzindo também a latência;
  - Simplificação do hardware em consequência da não fragmentação e de não necessitar de recalcular a checksum.
- Routing mais eficiente:
  - O Routing em IPv4 implicava gerir uma lista significativa de blocos por cada AS (Autonomous System, ou seja, as atribuições de cada operador). Já em IPv6, como as atribuições são bastante grandes, cada operador tem tipicamente um único bloco;
  - Sem NAT não existe necessidade de tradução de endereços públicos/privados.

O IPv4 será descontinuado num futuro próximo, ficando apenas a funcionar o IPv6.

IPv6 para tótós

O protocolo IPv6 foi originalmente publicado no RFC1883 [7], e actualizado pelo RFC2460 [8], entre outros. Aqui fica o resumo que qualquer profissional de informática deve saber.

Consoante o tipo de ligação disponível, há várias alternativas para acesso aos mundos IPv4 e IPv6:

- Apenas IPv4 nativo:
  - É possível ligar a IPv6 utilizando Túneis [9] (6in4, 6to4/Teredo, 6RD, etc.).
- IPv4 + IPv6 nativos (chamado Dual Stack)
  - Acesso directo sem túneis;
  - Acesso IPv4 pode eventualmente utilizar NAT para partilhar um endereço público com vários dispositivos com endereços IPv4 privados.
- Apenas IPv6 nativo
  - É possível ligar a IPv4 utilizando Túneis e outros métodos (NAT64/DNS64, 464XLAT, DS-Lite, etc.)
  - Muito utilizado na Ásia onde não é viável facultar endereços IPv4 públicos aos utilizadores;
  - A título de curiosidade, a infraestrutura interna do Facebook tem exclusivamente endereçamento IPv6, só os frontends suportam IPv4.

```
LINUX

User@Example.Andy.PT:/PaP # ping FACEBOOK.COM
PING FACEBOOK.COM (173.252.89.132) 56(84) bytes of data.
[...]

User@Example.Andy.PT:/PaP # ping6 FACEBOOK.COM
PING FACEBOOK.COM(edge-star-mini6-shv-17-prn1.facebook.com) 56 data bytes
[...]

WINDOWS

D:\PaP> PING /4 FACEBOOK.COM
Pinging FACEBOOK.COM [66.220.146.36] with 32 bytes of data:
[...]

D:\PaP> PING /6 FACEBOOK.COM
Pinging FACEBOOK.COM [2a03:2880:2130:cf05:face:b00c:0:1] with 32 bytes of data:
[...]
```

Formato dos endereços IPv6:

- 128 bits;

IPV6 PARA PROGRAMADORES

- 32 dígitos hexadecimais (0-9, A-F), case insensitive;
- 8 grupos de 4 dígitos hexadecimais usando ":" como separador;
- Em cada grupo, os "0" à esquerda podem ser omitidos;
- "::" significa que todos os dígitos num ou mais grupos seguidos são "0". Só pode ser utilizado uma vez;
- "/" define o scope da subnet - usa dígitos em decimal (0 a 128);
- "%" define a interface física por onde passa a comunicação;
- Os endereços podem ser envolvidos em parêntesis rectos se necessário, em particular quando se usam números das portas.

O mesmo endereço IPv6 pode ser escrito de várias formas. Aqui fica um exemplo (os espaços estão presentes apenas para facilitar a leitura e não devem ser utilizados):

- DNS: **FCCN.PT (registo AAAA)**
- Endereço: **2001:0690:0A00:1036:1113:0000:0000:0247/128**
- Reduzido: **2001: 690: A00:1036:1113: 0: 0: 247/128**
- Canónico: **2001: 690: A00:1036:1113:: 247**
- Alocação da rede: **2001: 690::/29**
- Com um interface: **2001: 690:A00:1036:1113::247%21**
- Com uma porta: **[2001:690:A00:1036:1113::247]:80** (HTTP)

Também o PTR (pointer record - usado para reverse DNS, ou seja, para obter o endereço DNS a partir do endereço de IP) muda em IPv6:

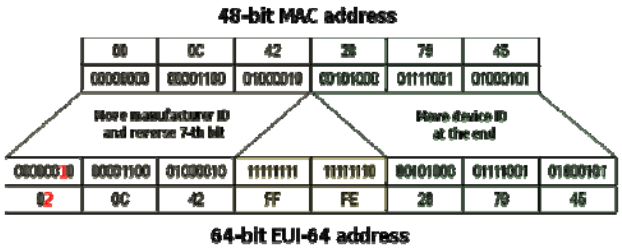
- PTR: **7.4.2.0.0.0.0.0.0.0.0.0.3.1.1.1.6.3.0.1.0.0.a.0.0.9.6.0.1.0.0.2.ip6.arpa. IN PTR FCCN.PT.**

Em IPv4 os operadores entregam normalmente um endereço público nas casas/empresas, mas com IPv6 é entregue uma rede inteira. O mínimo é um "/64", mas alguns operadores entregam uma gama maior, como "/60" ou "/56". Os clientes podem assim gerir as suas subnets como entenderem:

À semelhança de IPv4, existem alguns endereços especiais em IPv6. Por exemplo, o "::1" é equivalente ao "127.0.0.1" ou "LOCALHOST":

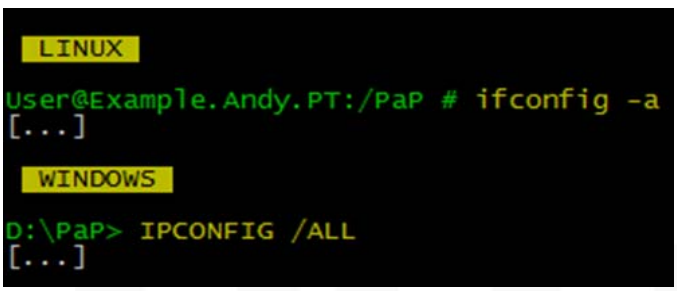
Addresses	Range	Scope
Loopback	::1	host
Link Local	fe80::/10	link
Unique Local	fc00::/7	global
Global Unicast	2000::/3	global
6to4	2002::/16	global
Multicast	ff00::/8	variable
Teredo	2001::/32	global

Em IPv4, para um dispositivo obter um endereço, utiliza DHCP (ou alternativamente configura o endereço IPv4 manualmente). Em IPv6 há várias formas de obter endereços, a maioria sem necessidade de haver um servidor que forneça endereços (Neighbour Discovery Protocol, DHCPv6, manualmente, etc.). O método mais usado é o de detectar pelos vizinhos qual o endereço público da rede (primeiros 64 bits), e utilizando o MAC Address do dispositivo para preencher os últimos 64 bits:



Isto pode, no entanto, ser considerado uma falta de segurança, já que pelo MAC Address é possível obter a marca/modelo do dispositivo. No limite, seria possível detectar que o mesmo dispositivo esteve presente em redes diferentes em locais diferentes (e assim gerar um perfil da utilização online da pessoa). Por esse motivo, foram criadas as "Privacy Extensions for Stateless Address Autoconfiguration in IPv6" [10], que em vez do MAC Address utilizam dígitos aleatórios, que substituem regularmente (e.g. a cada hora) para aumentar a privacidade.

Tipicamente, cada interface de rede tem vários endereços IPv6: o endereço Link-Local (baseado no MAC Address), o endereço de auto-configuração (também baseado no MAC Address), o endereço variável com base nas Privacy Extensions, entre outros. Como habitual, os endereços podem ser consultados usando:



Além dos endereços IPv6, toda a infraestrutura de nomes DNS funciona com IPv6 com pequenas alterações:

- IPv4: Um endereço é guardado usando um registo do tipo "A";
- IPv6: Um endereço é guardado usando um registo do tipo "AAAA" (lê-se "quad-A");
- Os registos do tipo "CNAME" ou "MX" continuam a funcionar em ambos, já que apenas apontam textualmente para outros registos do tipo "A" ou "AAAA";
- Os registos de SPF (tipo "TXT") suportam IPv6 usando o prefixo "ip6:";
- Os registos de reverse DNS (tipo "PTR") baseiam-se em "IP6.ARPA".

Ao fazer um pedido pelo nome, um dispositivo dual-stack tipicamente tentará resolver o endereço para IPv6 primeiro, pedindo o registo "AAAA", e apenas se não existir tentará pedir o registo "A". Tentará ligar-se primeiro ao endereço IPv6. Se falhar (timeout, etc.), tentará ao fim de alguns segundos o endereço IPv4. Caso haja uma demora significativa a estabelecer a ligação, pode ser sinal de haver um problema com a ligação IPv6 que teve failover para IPv4.

Em IPv4 existe o TCP e o UDP, ambos com 65535 portas (de 1 a 65535) que estão atribuídas a serviços. Um exemplo é a porta 80 utilizada pelo protocolo HTTP. Em IPv6 existe o TCPv6 e o UDPv6, em tudo semelhantes (excepto nos endereços), e cujas portas mantêm a mesma numeração e funcionalidade. Convém apenas salientar que, para evitar ambiguidades, é necessário envolver os endereços com parêntesis rectos quando se indica o número da portas.

- IPv4: **123.123.123.123:80** (HTTP)
- IPv6: **[456:789::ABCD]:80** (HTTP)

Existe muito mais informação online sobre IPv6. Recomenda-se a leitura da página da Wikipédia de IPv6 [11], cujo conteúdo é bastante acessível.

Recomendações para programadores

A maioria dos servidores web (IIS, Apache, NGINX, etc.) suporta IPv6 de forma transparente. Aqui é necessário apenas garantir que estão *listening* em IPv6 e que eventuais configurações (e.g. listas de acessos por IPv6, etc.) estão devidamente preenchidas e testadas.

Tirando as situações listadas de seguida, um programador web pode estar descansado que tudo funcionará em IPv6 da mesma forma que em IPv4.

A maioria dos CMS/CRM/ERP pode registar os endereços IPv4 que acederam ao respectivo site, mas só recentemente estes começaram a ser compatíveis com endereços IPv6. Afinal o que é necessário para armazenar endereços numa base de dados (relacional ou outra) ?

- Um endereço IPv4 necessita de 15 caracteres de armazenamento;
- Um endereço IPv6 necessita de 39 (endereço) + 4 (subnet) + N (interface). Ou seja, pelo menos 48 caracteres são recomendados;
- O mesmo campo pode ser usado para armazenar IPv4 e IPv6 (e.g. distinguir detectando se existe ":", o que implica ser IPv6, ou colocando um prefixo que diga qual a versão, etc.);
- Não é recomendado armazenar em formato binário ou numérico;
- Podem ser armazenados em formato canónico (reduzido) ou expandido (incluindo todos os zeros, com ou sem os ":");
- É necessário escolher se são armazenados em maiúsculas ou minúsculas, sendo também recomendado que a collation do respectivo campo seja case-insensitive;
- Não é recomendado armazenar apenas os primeiros 64 bits do cliente, já que múltiplos clientes podem estar a partilhar a mesma ligação, tendo apenas os últimos 64 bits como diferença.

Na gestão e/ou validação de acessos, é habitual garantir que pedidos sucessivos do mesmo utilizador originam do mesmo endereço de IP. Algumas implementações aceitam que o IPv4 mude, desde que na mesma subnet (aceitar qualquer número na última parte do endereço, ou seja, o endereço IPv4 pertencer à mesma "/24"). Já em IPv6 as coisas são bem diferentes: com as "Privacy Extensions" o endereço IPv6 dum utilizador muda regularmente (pelo menos os últimos 64 bits). Assim, a recomendação é validar usando apenas os primeiros 64 bits do endereço (parte da rede), ignorando alterações aos últimos 64 bits (parte interna do cliente). Em alguns casos, pode detectar-se que o cliente tem atribuída uma rede maior (e.g. "/60") e ser necessário expandir a tolerância às respectivas subnets. Em qualquer caso, deve sempre haver um token identificador do cliente (e.g. cookie), porque tanto em IPv4 como IPv6 é possível encontrar dois clientes que partilham o mesmo endereço.

Este exemplo mostra o endereço e a versão de IP do cliente numa página web (nota: foi utilizado PHP mas noutras linguagens o código será semelhante):

```
<?php
    $Andy_IPAddress = StrToUpper(Trim($_SERVER
                                    ["REMOTE_ADDR"]));

    If (PReg_Match("/\d+\.\d+\.\d+/",
                                    $Andy_IPAddress) )
    {
```

```
        $Andy_IPVersion = "IPv4";
    }
    Else If (PReg_Match("/[0-9A-F]+\:[0-9A-f\:\.]+/
                            i", $Andy_IPAddress) )
    {
        $Andy_IPVersion = "IPv6";
    }
    Else
    {
        $Andy_IPVersion = "?";
    }
    Print ("\tYour IP Address is: <STRONG>" .
              $Andy_IPAddress . "</STRONG><BR>\n");
    Print ("\tYour IP Version is: <STRONG>" .
              $Andy_IPVersion . "</STRONG><BR>\n");
?>
```

E se tentarmos programar uma ligação a uma máquina remota, quais as recomendações?

- Sempre que possível usar endereços DNS e nunca endereços IPv4 ou IPv6. Isto garante que o sistema operativo decidirá qual a melhor forma de efectuar a ligação;
- Nos casos em que queremos reduzir o tempo despendido com a consulta DNS (e.g. ligação dum servidor web a uma base de dados), pode ser realmente ser necessário colocar o endereço. O recomendado é colocar o endereço IPv6, já que a latência é normalmente menor;
- Em algumas situações raras, pode ser útil incluir a interface no endereço, para evitar que o sistema operativo demore a escolher por qual deve fazer a ligação. No entanto, uma mudança de hardware poderá implicar alterações às configurações, pelo que não é recomendado.

Se uma aplicação ou página web necessitar realmente de obter um endereço IPv4 ou IPv6 (registos "A" ou "AAAA") a partir dum endereço DNS, pode usar funções de consulta DNS. Todas as linguagens mais utilizadas já suportam IPv6 (nota: estas funções variam muito de linguagem para linguagem). Para detectar se existe um determinado registo DNS:

```
<?php
   $Andy_DNS_NAME = "ANDY.PT"; //Don't forget the
                                    "." in the end!

   $Andy_Exists_A = (CheckDNSrr($Andy_DNS_NAME,
                            "A" ) ? "Yes" : "No");
   $Andy_Exists_AAAA = (CheckDNSrr($Andy_DNS_NAME,
                           "AAA") ? "Yes" : "No");

   Print ("\t" . $Andy_DNS_NAME . " A record
              exists? <STRONG>" . $Andy_Exists_A .
                            "</STRONG><BR/>\n");
   Print ("\t" . $Andy_DNS_NAME - " AAAA Record
         exists? <STRONG>" . $Andy_Exists_AAAA .
                            "</STRONG><BR/>\n");
?>
```

Obter um registo "AAAA" não é muito diferente de obter um registo "A":

```
<?php
   $Andy_DNS_NAME = "ANDY.PT"; //Don't forget the
                                    "." in the end!
 $Andy_Exists_A = DNS_Get_Record,( $Andy_DNS_NAME
                                          DNS_A);
 $Andy_Exists_AAAA = DNS_Get_Record
                        ( $Andy_DNS_NAME DNS_AAAA);

   Print_R ($Andy_DNS_Get_A);
   Print_R ($Andy_DNS_Get_AAAA);
?>
```

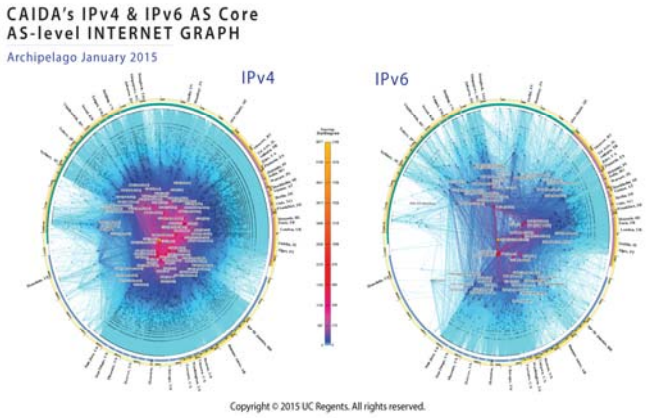
O resultado neste caso é:

```
Array
  {
    [0] => Array
        {
            [host] => FCCN.PT
            [class] => IN
            [ttl] => 3600
            [type] => A
            [ip] => 193.137.196.247
        }
  }
```

```
Array
  {
    [0] => Array
        {
            [host] => FCCN.PT
            [class] => IN
            [ttl] => 3600
            [type] => AAAA
            [ipv6] => 2001:690:a00:1036:1113::247
        }
  }
```

Anexos

Diagrama do routing mundial entre os principais AS (Autonomous Systems) da Internet (em IPv4 e IPv6):



Fonte: https://www.caida.org/research/topology/as_core_network/2015/ (2015-01)

Referências e links

Referências presentes no texto:

- [1] https://en.wikipedia.org/wiki/Internet
- [2] https://en.wikipedia.org/wiki/Internet_Protocol_Suite
- [3] https://en.wikipedia.org/wiki/Transmission_Control_Protocol
- [4] https://en.wikipedia.org/wiki/Network_address_translation
- [5] https://en.wikipedia.org/wiki/Maximum_transmission_unit
- [6] https://en.wikipedia.org/wiki/Path_MTU_Discovery
- [7] https://tools.ietf.org/html/rfc1883 (IPv6, RFC inicial)
- [8] https://tools.ietf.org/html/rfc2460 (IPv6, RFC actual)
- [9] https://en.wikipedia.org/wiki/IPv6_transition_mechanism
- [10] https://tools.ietf.org/html/rfc4941 ("Privacy Extensions for Stateless Address Autoconfiguration in IPv6")
- [11] https://en.wikipedia.org/wiki/IPv6

Dinamizadores em Portugal:

- Internet Society (Portugal Chapter): http://ISOC.PT
- FCCN/FCT: http://FCCN.PT/en/academic-network/ipv6
- Associação DNS.PT: http://DNS.PT
- Comunidade IPv6 Portugal: http://Facebook.COM/groups/IPv6.PT
- Comunidade IoT Portugal: http://Facebook.COM/groups/IoTPortugal

Dinamizadores internacionais:

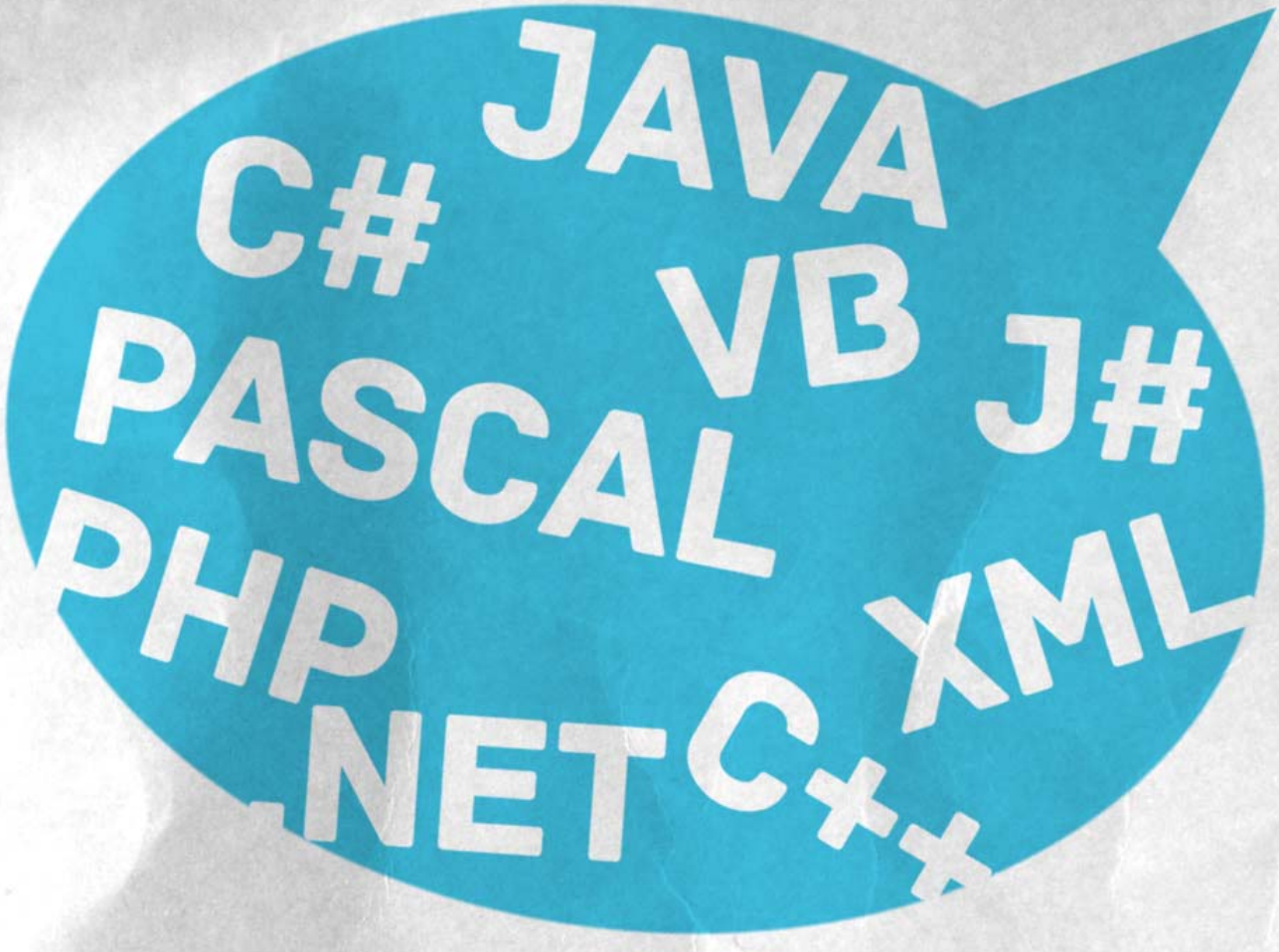
- Internet Society: http://ISOC.ORG
- RIPE NCC:
  - https://www.ripe.net/publications/ipv6-info-centre
  - https://www.ripe.net/support/training/courses/ipv6
  - https://www.ripe.net/support/training/courses/advanced-ipv6
- http://6deploy.eu
- Grupo IPv6 Global no Facebook: http://Facebook.COM/groups/2234775539

AUTOR

Escrito por André Melancia

Programador/DBA/Formador independente, Voluntário, orador e/ou participante regular nas comunidades IoT Portugal, IPv6 Portugal, DNSSec Portugal, ISOC.PT, SQLPort, NetPonto, Office365PT, SharePointPT, PTXug, PHPLx, etc.

Mais informações e contactos em http://Andy.PT



ENTÃO, SÓ FALAS EM CÓDIGO?

TEMOS O REMÉDIO CERTO PARA TI!

portugal-a-programar.pt

A MAIOR COMUNIDADE PORTUGUESA DE PROGRAMAÇÃO, APARECE!

ELECTRÓNICA

Aquisição de dados via TCP/IP com Genuino (Arduino)

AQUISIÇÃO DE DADOS VIA TCP/IP COM GENUINO (ARDUINO)

Introdução

Recentemente a tão conhecida marca Arduino , fundada por Massimo Banzi, David Cuartielles, Tom Igoe e David Mellis e toda uma comunidade, sofreu uma mudança de nome, para os produtos destinados a outros mercados fora dos EUA, passando a usar o nome Arduino apenas nos EUA e o nome Genuino, em todos os restantes mercados. Falo em marca, pois não se refere apenas a uma board, mas a toda uma "marca" de circuitos baseados em microcontroladores e projectos com base numa mesma filosofia de open-hardware. Não me alongando mais sobre o tema, esta mudança teve origem numa questão legal, que é muito bem apresentada por Maximo Banzi, no keynote que apresentou na Maker Fair e pode ser visto no youtube. Assim sendo, de agora avante, neste artigo, o circuito anteriormente conhecido por Arduino, será designado por Genuino.

Passemos agora àquilo que é realmente interessante: comunicar com o Genuino, receber dados e transmitir dados! Recentemente estive a trabalhar num projecto, que começou como trabalho e acabou dando origem a um fork para um hobbie, onde quis receber dados, de um amplo conjunto de sensores, ligados ao microcontrolador e transmiti-los de seguida, usando a ligação de rede existente para essa transmissão, usando a pilha TCP/IP v4, para transmitir dados bidireccionalmente, entre o microcontrolador e um dispositivo ligado à rede. Neste caso concreto foi utilizada uma ligação por cabo, apesar de ser relativamente simples fazer o mesmo com um circuito ethernet, por exemplo um ESP8266-01, ou qualquer outro que suporte a norma 802.11b, quer comunique com o Genuino por UART, quer comunique usando um outro protocolo.

O problema

Neste caso concreto, pretendia obter a informação sobre a humidade relativa do ar, a temperatura e a luminosidade num determinado espaço e de seguida transmitir essa mesma informação via rede.

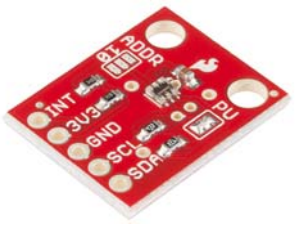
> ***"Um problema claramente apresentado, está meio resolvido." (Charles F. Kettering)***

Posto o problema, claramente definidos os objectivos, como seria de esperar, existe uma pesquisa sobre que sensores usar paca cada uma das funções e como o por a comunicar com a rede.

O hardware

Neste projecto concreto, os sensores usados foram um sensor DHT-11, conjuntamente com a respectiva resistência pull-up de 4.7k, na linha de dados, ligado ao Genuino. Este sensor oferece a capacidade de ler a temperatura do ar e a humidade relativa, é amplamente suportado pela comunidade e existe uma excelente biblioteca de suporte para o mesmo, a lib dht11.h disponível no github no repositório da Adafruit.

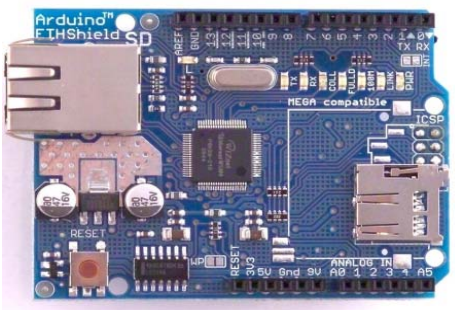
Para obter a informação da luminosidade, existiam duas possibilidades, logo à primeira vista. Uma seria usar um foto-resistor, a outra usar um circuito mais complexo como o TSL2561 da Sparkfun, que comunica por I2C com o microcontrolador e oferece uma série de funcionalidades interessantes que um fotoresistor não disponibiliza, nomeadamente a capacidade de ler tanto a luminosidade visível como a luminosidade infra-vermelha, aproximando os valores lidos, daqueles que o olho humano percepciona. Tal como o sensor DHT-11, o TSL2561 tem um amplo suporte de toda a comunidade e bibliotecas que facilitam a sua utilização.



TSL2561

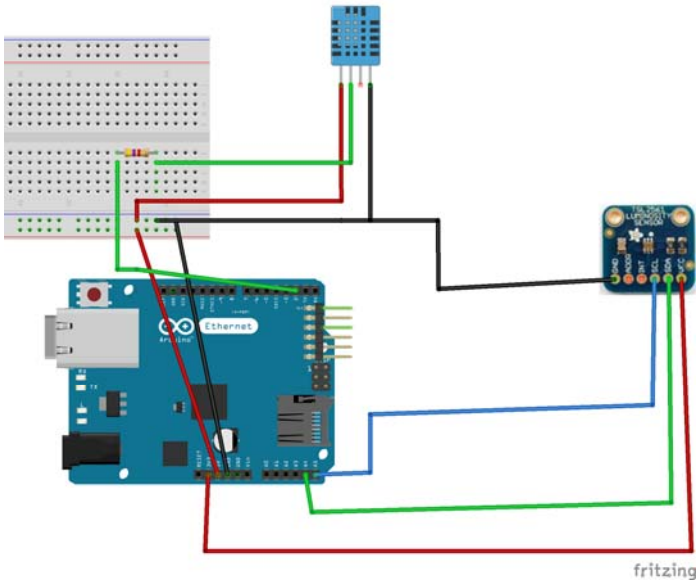
DHT-11

Escolhidos os sensores e a placa Genuino, no caso um Genuino Uno, restou a escolha do circuito ethernet a utilizar, neste caso ethernet shield w5100, com leitor de cartões micro-sd e stacking headers, para ficar acoplado mesmo por cima do Genuino.



Ligações

Escolhido o hardware, é necessário proceder às ligações antes de passarmos à escrita do código destinado a ser executado pelo microcontrolador ATMega328. Para este efeito e uma vez que não se trata de um circuito definitivo, mas apenas um circuito de teste, as protoboards, como se pode ver na imagem abaixo, são bastante uteis e facilitam as ligações.



A fim de facilitar as ligações do circuito, na tabela abaixo encontram-se os sensores, indicando o número de cada pino e o pino a que é ligado no circuito ethernet do Genuino.

DHT11		Genuino
Pino do sensor	Componente	Pino do Genuino
1		5vdc
2	Resistência 4.7k	D2
3		
4		Gnd
TSL2561		Genuino
1-SDA		A4
2-SCL		A5
3-GND		GND
4-3v3		3v3
5-INIT		

As células em branco foram deixadas propositadamente uma vez que nos respectivos locais, não é feita nenhuma ligação, quer seja a um componente intermédio, quer seja ao Genuino.

Programação do Genuino

Chegados a esta parte, está na hora de programar o Genuino de forma a comunicar por rede ethernet, usando o conjunto de protocolos TCP/IP. Felizmente uma parte substancial do trabalho, já vem "pré feito" pela biblioteca ethernet.h, que nos disponibiliza uma maior abstracção do hardware propriamente dito, deixando-nos livres para a programação da aplicação que será executada. No entanto continua a ser necessário executar algumas tarefas como a definição de um mac address, a colocação em modo cliente DHCP, caso tenhamos um servidor DHCP na rede, ou a definição de um endereço ip v4, bem como a respectiva máscara de sub-rede e gateway. Desta biblioteca iremos usar maioritariamente o método write, da classe server, para enviarmos dados para o nosso cliente TCP, bem como o método read, para lermos instruções transmitidas pelo servidor.

O código não é complexo e é quase auto-explicativo, não me alongando na sua explicação. De forma muito resumida, inicia a interface ethernet, fica a aguardar por uma conexão e quando a mesma está aberta, aguarda um comando, que executa e devolve o seu resultado, em texto simples. Recomendo a leitura da documentação oficial da biblioteca, para quem pretenda aprofundar os conceitos e conhecimentos.

```
//thanks to the sparkfun team, for the li
//braries, all credits for their work due to
//them.
#include "DHT.h"
#include <SparkFunTSL2561.h>
#include <Wire.h>
#include <SPI.h>
#include <Ethernet.h>

#define DHTPIN1 2     // define o pino de dados
                      //do sensor, neste caso 2
#define DHTTYPE DHT11   // DHT 11 (basta uma
                    //constante para dois sensores)
SFE_TSL2561 light;
boolean gain;   //definição do ganho, 0 = X1,
                //1 = X16;
unsigned int ms;
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE,
                                          0xED };
IPAddress ip(192,168,1, 177);
IPAddress gateway(192,168,1, 1);
IPAddress subnet(255, 255, 255, 0);

DHT dht1(DHTPIN1, DHTTYPE);

EthernetServer server(77);
boolean alreadyConnected = false;

String commandString;

void setup() {
  dht1.begin();
  light.begin();
  unsigned char ID;
  if (light.getID(ID))
  {}
  else
  { byte error = light.getError(); }
  gain = 0;
  unsigned char time = 2;
```

AQUISIÇÃO DE DADOS VIA TCP/IP COM GENUINO (ARDUINO)

```
  light.setTiming(gain,time,ms);
  light.setPowerUp();
  pinMode(ledPin, OUTPUT);
  Ethernet.begin(mac, ip, gateway, subnet);
  server.begin();
  Serial.begin(9600);
   while (!Serial) {;}
  Serial.print("Socket Server:");
  Serial.println(Ethernet.localIP());
}

void loop() {
  EthernetClient client = server.available();
  if (client) {
    if (!alreadyConnected) {
      client.flush();
      commandString = "";
      server.println("--> insira um dos comandos
                                    disponiveis!");
      alreadyConnected = true;
    }
    while (client.available()) {
     char newChar = client.read();
     if (newChar == 0x0D)
     { processCommand(commandString);}
      else { commandString += newChar; }
    }
  }
}

void processCommand(String command)
{
  //dht11-1
    if (command.indexOf("dht1") > -1)
    {
      delay(1000);
      float h1 = dht1.readHumidity();
      float t1 = dht1.readTemperature();
      if (isnan(h1) || isnan(t1))
      { return; }
      server.print("Humidade ");
      server.print(String (h1));
      server.print("; Temperatura ");
      server.println(String (t1));
      commandString = "";
      return;
  }
  //light
    if (command.indexOf("light") > -1)
    {
      delay(200);
      unsigned int data0, data1;
      if (light.getData(data0,data1))
        {
            double lux;
            boolean good;
            //calculo de Lux
            good = light.getLux
                        (gain,ms,data0,data1,lux);
            server.print("luminosidade em lux ");
            server.println(lux);
        }
      commandString = "";
      return;
    }
  commandString = "";
  instructions();
}
void instructions()
   server.println("Comando não reconhecido");
   server.println("Use um dos seguintes
                                        comantos:");
   server.println("* dht1, para obter a humidade e
                                  temperatura do ar");
   server.println("* light, para obter a informação
                                    da luminosidade");
}
//to my late night owl angel, thanks for the wisdom
//inspiration and company,
//Stephan|B|
```

Conclusão

Uma vez feito o upload do código, para o Genuino, basta ligar um cabo de rede e um cliente simples ou usar a nossa própria aplicação para comunicar por sockets com o Genuino, enviando a instrução que pretendemos e recebendo o resultado da mesma. Neste exemplo usei o putty, para simplificar, apenas defini a porta como 77, pois era uma das portas disponíveis. O objectivo ao usar sockets neste exemplo, prendeu-se com a simplicidade com que se comunica com o dispositivo, bem como com o leque de possibilidades que deixa em aberto para o desenvolvimento de aplicações que interajam com o circuito, como é o caso de aplicações mobile, feitas por exemplo com C# e Xamarin, usando o componente “sockets-for-pcl”, que é bastante simples de usar e bastante intuitivo.

> ***“Se procuras resultados distintos, não faças sempre o mesmo!” (Albert Einstein)***

AUTOR

Escrito por António C. Santos

Com uma enorme paixão por tecnologia, autodidacta desde tenra idade, cresceu com o ZX Spectrum. Tem vasta experiência em implementação e integração de sistemas ERP, CRM, ERM, BI e desenvolvimento de software por medida nas mais diversas linguagens. Diplomado do Curso de Especialização Tecnológica em Tecnologias e Programação de Sistemas de Informação pela ESTG-IPVC. Membro da Comunidade Portugal-a-Programar desde Agosto de 2007, é também membro da Sahana Software Foundation, onde é Programador Voluntário. Neste momento é aluno no Instituto Politécnico de Viana do Castelo, na Escola Superior de Tecnologia e Gestão no curso de Licenciatura em Engenharia Informática e Xamarin Student Partner nesta mesma escola Twitter:@apocsantos

COLUNAS

Kernel Panic - Cross-Platform - “A soma de todas as mudanças”

Cross-Platform - “A soma de todas as mudanças”

Cada vez mais ouvimos falar em cross-platform! Não é algo novo, é algo já antigo em verdade se diga! Se nos referirmos ao cross-platform como a capacidade de compilar e correr o mesmo código em múltiplas plataformas de hardware poderemos recuar tanto quanto a criação da linguagem C, por Dennis Ritchie e Ken Thompson em 1972, quando trabalhavam nos Bell Labs. De forma brilhante, estes senhores decidiram escrever uma linguagem que permitisse compilar o código e executá-lo independentemente do hardware. Inicialmente era um DEC PDP-7, posteriormente um DEC PDP-11, tendo a linguagem sido utilizada para escrever maior parte do kernel do sistema operativo Unix.

Já nessa altura se falava e pensava em cross-platform, a um nível mais de hardware do que de sistema operativo, tendo como objectivo a redução do tempo consumido na portabilidade do código para novas plataformas de hardware e / ou software.

Durante anos, pouco se falou sobre este conceito. Esta mudança foi principalmente da forma de pensar, havendo claramente quem pensasse e defendesse de forma convicta que o código escrito para uma plataforma não deveria correr noutra plataforma diferente, quer fosse hardware ou software (sistema operativo), diferente. De um lado os PC-Compatíveis com software escrito para arquitectura x86. De outro os computadores a Apple que corriam software escrito para uma arquitectura de hardware diferente baseada em processadores da Motorola, desde o 68000 até aos PowerPC, várias gerações de processadores e sistemas operativos que não corriam em PC-Compatíveis, formando ecossistemas fechados e isolados. Corria o pensamento não propriamente coincidente com o “cross-platform”, onde o código escrito para uma plataforma, apenas correria naquela plataforma e caso tivesse de ser portado, seria reescrito na íntegra, consumindo tempo e recursos, nem sempre disponíveis, limitando tanto os utilizadores, como os criadores de software.

Apesar de ter começado a soprar o “vento da mudança” do cross-platform em 1972, o “platform-specific”, manteve a “supremacia” desde então, tendo soprado de novo com mais alguma “força” em meados de 1995. A esta altura a Sun Microsystems disponibilizou a primeira implementação da linguagem de programação Java, o Java 1.0, sob o lema “Write Once, Run Anywhere” (WORA), traduzido “Escreve uma vez, corre em qualquer parte”.

Sem custos de execução, permitia correr o código nas plataformas mais populares da altura. Relativamente segura (ao que era conhecido à data de publicação da linguagem), permitia entre outras funcionalidades restrições de acesso a redes e a ficheiros. Na época uma esmagadora maioria dos navegadores web, incorporaram a capacidade de executar applets java dentro de paginas web, dando assim uma ainda maior popularidade a esta linguagem e ao “movimento cross-platform”, que não sendo um movimento formal, era uma ideia, um conceito, um objectivo, uma filosofia, que se mantinha viva, ainda que discretamente.

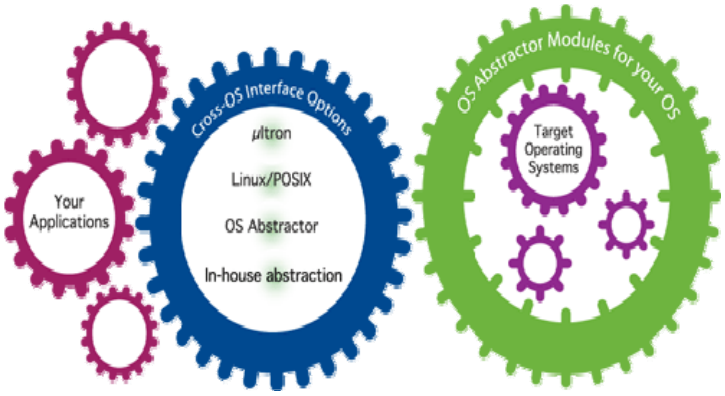
No entanto, esta linguagem veio “perder algum ímpeto” no caminho do cross-platform em 1997, aquando de uma disputa legal entre a Sun Microsystems e a Microsoft Corporation, em que a Sun Microsystems, detentora da linguagem, reclamava que a implementação da Microsoft não suportava RMI ou JNI e continha código “platform-specific”, (específico a plataformas) contendo funcionalidades das quais a Microsoft era proprietária. Esta disputa levou a um acordo em 2001 e ao facto do sistema operativo Windows deixar de trazer as ferramentas de execução de aplicações feitas em Java no seu pacote “package”. Este poderá ser considerado um dos revezes do cross-platform, uma vez que com esta tomada de posição, o Java, apesar de continuar a correr nas plataformas Microsoft, deixou de vir pré-instalado, o que leva a mais tarefas por parte do utilizador e como tal, menos adesão a esta linguagem e toda a filosofia subjacente.

Mas os tempos mudam, os ventos mudam, tudo muda, a indústria avança, a tecnologia melhora e o que outrora era ficção científica, passa a ser uma realidade. Por exemplo os smartphones! Certamente alguns se recordam do “Communicator” da série Star Trek. Um dispositivo que permitia comunicar de forma idêntica ao que agora conhecemos como telemóvel e até “smartphone”, mas que à data da transmissão televisiva da série, não era mais do que ficção científica. Ora bem, o advento dos dispositivos móveis, da computação móvel, dos chamados “smart devices”, trouxe mais plataformas de hardware e sistemas operativos para aquilo que já era um “mundo pouco cross-platform” com poucas plataformas! A grande variedade de sistemas operativos e arquitecturas de hardware destes dispositivos, cada um com os seus SDK’s, linguagens e API’s, dificultava ainda mais o conceito do cross-platform do “write once, run anywhere”, o conceito de Ritchie e Thompson, de 1972: escrever o código uma vez, compilar e correr em qualquer plataforma. No entanto esta ideia, este “movimento”, não tinha “falecido”. Apenas tinha encontrado os seus revezes, ganhando fôlego e balanço, fazendo “avanços à retaguarda” e avanços em frente, conforme o mundo evoluía.

Poderia citar aqui, uma série de ferramentas e players do desenvolvimento cross-platform, mas como se trata de um artigo de opinião e não de algo publicitário, nem de uma resenha histórica, damos um “salto”, para uns anos mais há frente e para umas quantas tecnologias “depois”, deixando apenas aqui a menção a um projecto que me cativou a atenção desde o seu inicio: o Mono. Um projecto “singular”, de código aberto, software livre, que pretendeu criar uma implementação livre e cross-platform da linguagem C# e da fra-

mework .NET da Microsoft, para que corresse em todas as plataformas de hardware e software, desde computadores x86 / IA64 com sistema operativo Windows, até outros dispositivos e sistemas operativos como GNU/Linux, BSD Mac OS X da Apple, Solaris e até nas consolas de jogos como o caso da PlayStation 3, Nintendo Wii e Xbox 360. Apesar de controverso dentro da própria comunidade open-source, conseguiu avançar, manter-se, existir e dar origem ao que agora muitos poderão conhecer como Xamarin, fornecendo ferramentas de desenvolvimento cross-platform para dispositivos móveis, que executem os sistemas operativos Windows Phone, Android e iOS.

Este foi um projecto que cativou a minha atenção desde o seu início, por permitir desenvolver para diversas plataformas e ser um projecto de código aberto, livre, deixando a liberdade criativa para os programadores fazerem o que fazem bem: *Criar novos mundos, melhores mundos, no código que escrevem*! Esta ideia, esta filosofia, cativou-me pelo que recomendo o leitor a explorar o projecto Mono e a filosofia do desenvolvimento cross-platform, como ela se encontra descrita em diversos artigos, sendo a parte mobile bem focada no artigo de Stephen Fluin, no artigo de Henning Heitkötte, entre outros.



Não sendo o objectivo deste artigo falar de ferramentas e dando o salto em frente, neste momento vivemos numa altura em que o desenvolvimento cross-platform parece estar num ritmo de aceleração quase vertiginoso, em que cada vez aparecem mais dispositivos e plataformas de hardware, mas todas elas já suportando os sistemas operativos existentes e de "main stream", como o caso do GNU/Linux, UNIX, Windows, Android.A Poucos são os fabricantes que ainda mantêm os seus sistemas operativos a correr apenas no hardware por si produzido, como o caso do iOS e do Mac OS. Estes sistemas já permitem execução de aplicações cross-platfrom, sendo que todos os sistemas operativos acima referidos, promovem uma cada vez maior abstracção entre o programador de aplicações e o hardware em que estas aplicações são executadas a um nível substancialmente mais baixo.

As mudanças, decorridas ao longo de décadas de avanços e recuos, de pontos altos e pontos baixos, de ideias contraditórias e personalidades divergentes, foram todas elas contribuindo e somando para que o desenvolvimento cross-platform fosse uma realidade! O desenvolvimento em que o programador, o "criador" de software, não tem de se preocupar com o hardware onde o seu programa vai ser executado, tão pouco com o sistema operativo onde este será executado, é de longe a maior vantagem desta "nova" realidade. Basta escolher uma boa ferramenta de desenvolvimento cross-platform, e tudo será resolvido! Não haverá necessidade de recompilar e reescrever código, para correr na plataforma A ou B, ou ter de fazer opções entre que plataformas suportar e não suportar, limitando tanto o programador, como o utilizador que por querer usar uma determinada aplicação teria de adquirir hardware e sistema operativo que a executasse em concreto.

A evolução da tecnologia, das ideias e do pensamento, levou a este somatório de acontecimentos que nos trouxe até aqui e é provável que num futuro próximo, nos leve bastante mais além, a um nível ainda mais simples desenvolvimento cross-platform, onde todas as aplicações serão "universais", executadas em todo o lado, sem uma dependência quase cega de um sistema operativo nem uma arquitectura de hardware. Tal como em 1972, em que dois carismáticos e brilhantes cientistas idealizaram e escreveram toda uma linguagem que permitia esse desenvolvimento e que de seguida os levou à reescrita de todo um sistema operativo, nessa mesma linguagem, para que "fossemos libertados" assim da dependência de um determinado hardware para executarem o seu software.

Até lá, programemos e façamos do mundo um lugar melhor, mais especial, escrevendo código!

Referências:

- Evaluating Cross-Platform Development Approaches for Cross-Platform Development in C++ (Syd Logan)

AUTOR

Escrito por António C. Santos

Com uma enorme paixão por tecnologia, autodidacta desde tenra idade, cresceu com o ZX Spectrum. Tem vasta experiência em implementação e integração de sistemas ERP, CRM, ERM, BI e desenvolvimento de software por medida nas mais diversas linguagens. Diplomado do Curso de Especialização Tecnológica em Tecnologias e Programação de Sistemas de Informação pela ESTG-IPVC. Membro da Comunidade Portugal-a-Programar desde Agosto de 2007, é também membro da Sahana Software Foundation, onde é Programador Voluntário. Neste momento é aluno no Instituto Politécnico de Viana do Castelo, na Escola Superior de Tecnologia e Gestão no curso de Licenciatura em Engenharia Informática e Xamarin Student Partner nesta mesma escola Twitter:@apocsantos

Media Partners da Revista PROGRAMAR

Análises

MySQL

Programação em Python: Fundamentos e Resolução de Problemas

MySQL

Título: MySQL

Autores: Frederico Tavares

Editora: FCA - Editora de Informática

Páginas: 248

ISBN: 978-972-722-803-4

Tive o prazer de receber e ler o livro MySQL de Frederico Tavares, Licenciado em Física pela Universidade de Coimbra e Mestre em Informática – ramo de Sistemas Distribuídos, Comunicações por Computador e Arquitectura de Computadores pela Universidade do Minho.

O SGBD MySQL afirma-se cada vez mais como o principal sistema de base de dados no que toca, pelo menos, ao mundo web. Confirmação disso é a utilização que os monstros empresariais lhe dão, como por exemplo o Facebook, LinkedIn, Twitter, Cisco, Ebay ou Amazon.

Frederico Tavares ensina-nos, de forma disciplinada e elegante, ao longo das 248 páginas, o que é, para que serve e como aplicar ao mundo real o propósito de ter uma base de dados. O livro aplica-se a todas as faixas etárias bem como a profissionais das tecnologias de informação e público em geral com interesse no ramo.

Sublinho a estrutura e organização do livro. Capítulos progressivos de aprendizagem que assentam em níveis básico, intermédio e difícil, diversos exemplos ilustrativos, que usam como ferramenta de auxílio o MySQL Workbench. Perfeito para quem se está a iniciar, perfeito para quem deseja adquirir mais conhecimentos.

A obra está dividida em 9 capítulos que consolidam a linha de ascendência em termos de complexidade:

- Instalação do MySQL
- Modelação e desenho da base de dados
- SQL para iniciantes
- Operações com tabelas
- Stored procedures e funções
- Cursors, triggers e erros
- Utilizadores e acessos
- Gestão do servidor
- Cópias de segurança

Após a leitura desta obra o leitor sentir-se-á apto ao pensamento crítico sobre o tema e terá ganho as competências fundamentais à delineação e criação da estrutura da base de dados MySQL.

Sentir-se-á igualmente capaz de reproduzir os seus conhecimentos noutros SGBD.

Os primeiros quatro capítulos são destinados a leitores cujo nível de conhecimento se assemelha ao básico ou nulo. O escritor tem o cuidado de abordar estes temas sobre 98 páginas, o que reflecte o cuidado e relevância que estes capítulos têm, forçando assim o leitor a adquirir boas práticas.

Os restantes capítulos dirigem-se ao público que procura maximizar tempo, esforço e desempenho das suas bases de dados e que já têm uma noção avançada de como trabalhar com o MySQL.

No último capítulo o autor escreve sobre as normas de segurança – nomeadamente de backup. É então abordado sobre 8 páginas a funcionalidade Slave e Master que muitos tendem a não conhecer e que assenta num paradigma de segurança e estabilidade de dados. Esta funcionalidade permite em tempo real manter um ou mais backups da base de dados principal (Master) em base de dados secundárias (denominadas slaves) sem qualquer intervenção do utilizador.

Por fim, aconselho a aquisição do livro pois o valor monetário pago converter-se-á em conhecimentos sólidos e análise crítica.

AUTOR

Escrito por Fábio Pinho

Programador entusiasta nas mais diversas linguagens, sendo PHP, .NET e Java (Android) as suas preferências.

Programação em Python: Fundamentos e Resolução de Problemas

Título: Programação em Python: Fundamentos e Resolução de Problemas

Autores: Ernesto Costa

Editora: FCA - Editora de Informática

Páginas: 632

ISBN: 978-972-722-816-4

O livro Programação em Python: Fundamentos e Resolução de Problemas, é de extrema utilidade para todos aqueles que desejem aprender não apenas a programar mas também os diversos paradigmas de programação, não necessitando de grandes conhecimentos prévios. O livro encontra-se estruturado de uma forma bastante organizada, simples e direccionada. Apreciei o facto de que o leitor é convidado a aprender primeiramente os conceitos teóricos necessários à compreensão das matérias apresentadas, acompanhando exercícios que proporcionam uma consolidação do conhecimento, agradável, ainda que pouco observável em obras técnicas como o livro em questão.

Ao ler o primeiro capítulo (introdução), compreendem-se conceitos fundamentais de arquitectura de computadores, funcionamento dos mesmos, linguagens e paradigmas de programação, além de noções de algoritmia importantes para uma sólida compreensão dos conteúdos da restante obra. Nota-se a atenção do autor para com o leitor, mantendo o texto sempre cativante, apelativo, motivador e atractivo, proporcionando desafios em forma de exercícios, que acabam por promover uma leitura mais atenta e interessada, até mesmo para os leitores menos "prováveis", pois programar nos dias de hoje pode ser considerado tão importante como saber ler e escrever o idioma nativo, de forma fluente.

Ao longo dos capítulos dois e três, abordam-se temas de igual importância como o caso de formas graficamente representadas e programação de interacções com o utilizador. Seguidamente é apresentada uma abordagem sólida, ainda que não aprofundada, aos objectos, começando por explicar os conceitos mais teóricos e aprofundando os conceitos mais práticos, como os tipos básicos de objectos, operações e precedências de operações, âmbito, métodos gerais e específicos, construtores e por fim, um tipo de objecto que merece alguma atenção especial: oTuplo (tuple) que sendo muito semelhantes às listas, têm no entanto especificidades concretas, muito úteis para armazenamento de dados em memória volátil.

No quarto capítulo, são apresentadas as instruções destrutivas e de controlo, de leitura e escrita. Mais uma vez se nota a atenção do autor na transmissão clara dos conceitos bem como na consolidação dos mesmos.

Com estes capítulos lidos, nota-se o aprofundar dos conteúdos, tendo por base os capítulos antecedentes, promovendo assim uma leitura progressiva e "ágil". Ao longo dos capítulos quinto e sexto, são aprofundados temas como instruções de controlo de fluxo e salto, instruções condicionais, bem como construtores, mutabilidade, cópia profunda, listas, cadeias, dicionários (particularmente úteis aquando da utilização da ferramenta Web2Py), árvores e outras estruturas de dados. Todos estes temas, são acompanhados de exercícios e culminam num teste de conhecimentos. Ao capítulo sétimo, dou-lhe o destaque pelo tema em causa, a saber: ficheiros. É um tema muitas vezes considerado complexo, mas de relevo e interesse para a programação, sendo que a apresentação deste tema no livro está simplesmente excelente, uma vez que é feita de forma simples e objectiva, sem "mistificações" desnecessárias, sendo de fácil compreensão.

> ***"Mais uma vez se nota a atenção do autor na transmissão clara dos conceitos bem como na consolidação dos mesmos."***

Tal como anteriormente, nos capítulos seguintes, são aprofundados os temas previamente abordados, como o caso dos objectos graficamente representados, sendo que a sua apresentação neste caso concreto se encontra de uma forma muito apelativa para o leitor, uma vez que apresenta uma utilização mais prática com imagens e objectos reais. Posteriormente chegados ao nono capitulo, o autor foca-se em aspectos mais complexos da programação, como o caso da recursividade, âmbito das variáveis e objectos, bem como programação orientada a objectos. Adopta uma aproximação clara, sólida e progressiva ao paradigma, à programação orientada a objectos com Python e seus aspectos mais avançados como herança, polimorfismo, encapsulamento, classes abstractas, entre outros, dispondo de exemplos práticos de grande utilidade e qualidade. Esta obra, consegue ainda abordar alguns aspectos da engenharia de software, ao

apresentar UML no exercício da "máquina multibanco".

Achei de particular interesse, tendo-me captado a atenção, o facto de o autor apresentar o tratamento e visualização de imagens provenientes de ficheiros. Isto é algo não muito comum num livro de Python, mas que se revela particularmente interessante, pois desperta interessa no leitor para temas que de outra forma, muito provavelmente acabariam por passar despercebidos. De igual modo podem ser úteis caso o utilizador pense dar continuidade aos exemplos de jogos que se encontram na parte final do livro, acrescentando por exemplo, uma imagem de fundo, de entre um vastíssimo leque de situações nas quais os conhecimentos transmitidos nesta obra, são de grande utilidade e valência, mesmo para quem pense que a programação se destina apenas a um público mais técnico e científico.

A título de exemplo apenas ilustrativo desta opinião pessoal, a linguagem Python é de extrema utilidade quando se trabalha com ferramentas de edição de imagem como o caso do GNU Image Manipulation Program (GIMP), para o qual é possível desenvolver toda uma série de plug-ins em Python. Creio poder considerar que um profissional da edição de imagem que utilize a ferramenta acima mencionada, após a leitura desta obra, sentir-se-á mais capaz de desenvolver código que lhe automatize tarefas e até acrescente funcionalidades à ferramenta.

No último dos capítulos o autor foca-se inteiramente nos aspectos relacionados com as interfaces gráficas de utilizador, algo que é essencial num programa de computador que execute mais tarefas do que simplesmente o tratamento de dados provenientes de suporte informático. A apresentação do tema, permite ao leitor uma consolidação dos temas anteriormente abordados, além de levantar interesse adicional e proporcionar os conhecimentos necessários ao desenvolvimento de aplicações amigas do utilizador e graficamente atractivas.

O livro dispõe ainda de um exemplo de uma animação e de um jogo simples, como é o jogo do galo, também conhecido por "tic-tac-toe", que apesar de ser "trivial" e impossível de vencer, caso os dois jogadores estejam ao mesmo nível de atenção e capacidade cognitiva, pois dada a simplicidade do jogo a probabilidade de empate entre jogadores em situação de igualdade de faculdades é de quase 100%.

Num registo mais pessoal, creio que o livro "cumpre mais do que aparentemente promete" numa primeira apreciação da capa. Não parece prometer tanto quanto "cumpre" quando se lê a obra de forma atenta, algo raro em literatura da especialidade, pelo que saúdo o autor com alegria! A obra no seu todo é de uma qualidade excepcional, numa linguagem extremamente acessível. O texto não é de todo "maçador", como os menos ávidos da leitura por vezes dizem ao adjectivar livros. A atenção é mantida de forma tão eficaz que ousaria opinar que ao ler esta obra, se está perante um ciclo de formação orientado por um formador e não de uma obra de consulta, como uma esmagadora maioria das obras tende a ser utilizada.

Realço o facto de nesta obra se encontrarem ilustrações a cores, algo que não é comum e foi com agrado que constatei. Independentemente do rigor técnico e científico da obra, que diga-se é elevado, o facto de dispor de algumas ilustrações mais relevantes a cor, ajuda a captar a atenção inicial dos mais desatentos. Da mesma forma gostaria de destacar os "*intermezzos*" disponíveis entre os vários capítulos que compõem esta obra, sendo excelentes complementos para o leitor e onde o autor aborda de forma, diria que mais "suave", algumas questões mais complexas e alguns mitos ou "pseudo-dogmas", chamemos-lhe assim, da programação para quem é principiante, como o facto de ser "todos os problemas terem apenas uma solução", a apresentada pelo autor, professor, formador, a pessoa que transmite o conhecimento. O autor desmistifica este facto num dos *intermezzos*, apresentando uma argumentação simples e igualmente sólida.

Em conclusão, é um livro que recomendo vivamente tanto a programadores, como a estudantes de programação, entusiastas e não entusiastas, em suma a qualquer pessoa que se interesse um pouco, por tecnologia e que esteja disposto a aprender um pouco mais! Como referi anteriormente nesta análise, o livro é de uma qualidade impar, adequando-se a todas as pessoas que se interessem pela programação.

A aprendizagem de programação, é por si só algo de grande importância, nos dias que correm. Não obstante a linguagem Python, quer pela maturidade atingida, quer pela quantidade cada vez maior de bibliotecas disponíveis, para as mais diversas funções, pela sua versatilidade parece-me ser uma linguagem que todos devem aprender.

AUTOR

Escrito por António C. Santos

Com uma enorme paixão por tecnologia, autodidacta desde tenra idade, cresceu com o ZX Spectrum. Tem vasta experiência em implementação e integração de sistemas ERP, CRM, ERM, BI e desenvolvimento de software por medida nas mais diversas linguagens. Diplomado do Curso de Especialização Tecnológica em Tecnologias e Programação de Sistemas de Informação pela ESTG-IPVC. Membro da Comunidade Portugal-a-Programar desde Agosto de 2007, é também membro da Sahana Software Foundation, onde é Programador Voluntário. Neste momento é aluno no Instituto Politécnico de Viana do Castelo, na Escola Superior de Tecnologia e Gestão no curso de Licenciatura em Engenharia Informática e Xamarin Student Partner nesta mesma escola Twitter:@apocsantos

No Code

Windows 10: As novidades da atualização de novembro (Build 10586 - Version 1511)

ECMAScript 2015: a nova versão do JavaScript

Raspberry Pi Zero – O novo Membro da Família!

Beta-i HACKATHON Lisbon has been Hacked

Xamarin 4.0 – Tudo o que precisa para criar aplicações móveis

Impressoras 3D – Escolhe a tua

Xamarin para Docentes e estudantes

HIGH TECH FASHION

Projecto em Destaque na Comunidade p@p: Portugol +

WINDOWS 10: AS NOVIDADES DA ATUALIZAÇÃO DE NOVEMBRO

(BUILD 10586 - VERSION 1511)

Introdução

A Microsoft iniciou recentemente a distribuição via **Windows Update,** da primeira grande atualização para o **Windows 10,** após o lançamento da **versão RTM** no passado mês de julho.

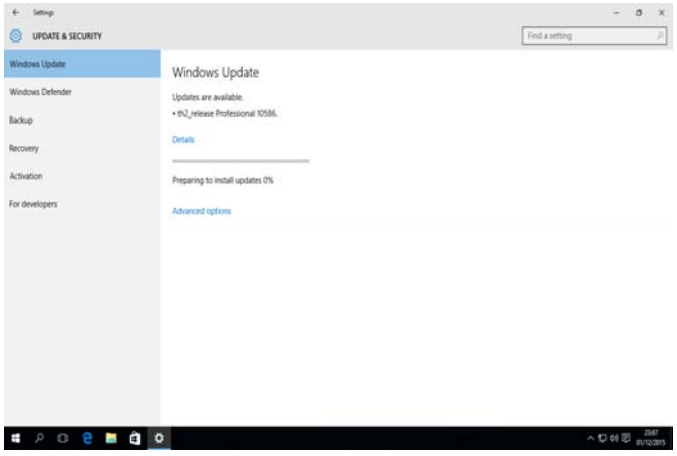
Ao longo dos últimos meses, a Microsoft trabalhou ativamente na correção de alguns problemas e através do **feedback** submetido pelos **Windows Insiders**, desenvolveu novas funcionalidades que pretendem melhorar tanto a performance, como a experiência de utilização do Windows 10.

Vejamos então em detalhe, quais as novas funcionalidades disponíveis para utilizadores domésticos e também as novidades para as empresas.

Instalação da atualização

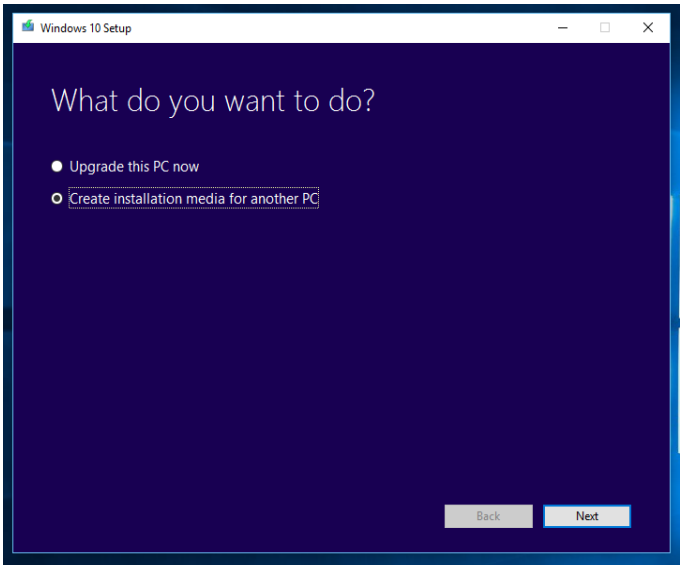
Tal como aconteceu com a versão RTM do Windows 10, a **Build 10586** está a ser distribuída de forma gradual via Windows Update. O processo de instalação é iniciado após o download dos bits necessários à atualização – cerca de **3 GB** - e demora dependendo do hardware, de 20 a 60 minutos a concluir.

Podemos verificar a existência da atualização, clicando no botão **Iniciar** e, em seguida **Definições** > **Atualização e segurança** > **Windows Update** > **Procurar atualizações**.



Windows Update

Em alternativa ao **Windows Update**, os utilizadores que queiram acelerar o processo de atualização, podem recorrer ao **MCT** (**Media Creation Tool**) que a Microsoft disponibiliza no site do Windows 10. O MCT já está atualizado com os **bits** da nova versão e para além do **"in-place" Update**, permite preparar uma **Pen USB** com os bits de instalação das versões **32** e **64 bits** e que podemos usar posteriormente para efetuar uma instalação limpa.



Microsoft Media Creation Tool

Algumas considerações sobre a atualização:

- Se a última atualização para o Windows 10 tiver sido há menos de **31 dias**, não será possível obter a atualização de novembro de imediato; isto irá permitir-lhe voltar à **versão anterior** do **Windows**, se assim o preferir. Depois de terem passado **31 dias**, o PC irá **automaticamente** fazer o **download** da **atualização de novembro**.

- Qualquer **atualização** instalada no PC **antes** da atualização de novembro já não será listada no **histórico** de atualizações.

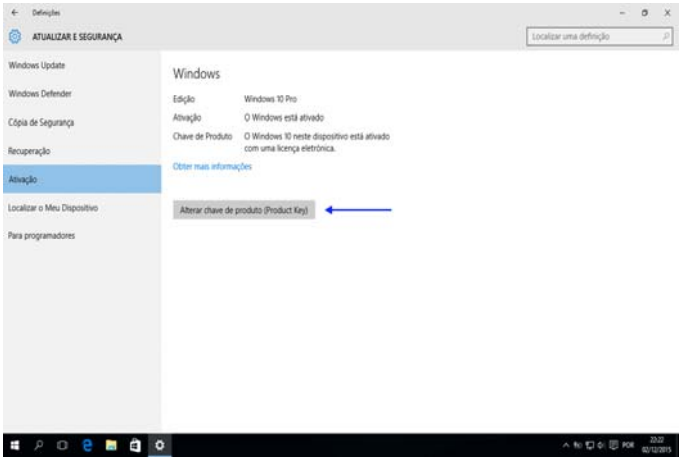
Novidades da versão

Ativação do Windows

O **Windows 10** introduziu um novo método de ativação – a **elegibilidade digital** - que não exige a introdução de uma chave de produto, ou seja, após a instalação da atualização gratuita, o Windows 10 é ativado automaticamente com base na **chave de produto genuína** do **Windows 7**,**8** ou **8.1**.

Na atualização de novembro, este método de ativação foi alvo de melhorias e passa a ser possível ativar o **Windows 10 (Versão 1511 ou superior)** através da introdução de alguns tipos de chaves de produto

do **Windows 7**, **Windows 8** e **Windows 8.1.**, após a instalação da atualização via Windows Update (caso a ativação não seja feita automaticamente) ou durante uma instalação limpa.



Alteração chave de Produto

Para fazermos a **ativação** do Windows 10, utilizando uma chave genuína destas versões, podemos proceder da seguinte forma:

- **Clicar** no **Iniciar**, em seguida, **clicar** em **Definições > Atualização e segurança > Ativação**.
- **Clicar** em **Alterar chave de produto** e introduzir a **chave de produto** de **25 carateres**.

Depois de concluído o processo de ativação, é então concedida a **elegibilidade digital** ao PC com base na chave genuína que introduzimos. Caso seja necessário fazer uma formatação ao dispositivo e uma vez que este foi ativado usando este novo método, não será necessário introduzir uma chave de produto durante a instalação para que o Windows seja de novo ativado neste equipamento.

Dos tipos de **chaves de produto** que podemos utilizar para ativação do Windows 10, não são suportados os seguintes tipos: **Licenciamento de Volume** que incluem as chaves **GVLK (Generic Volume Licensing Keys)** e **MAK (Multiple Activation Keys)** e também as chaves das versões **Enterprise** do **Windows 7**, **8** e **8.1**.

Menu Iniciar

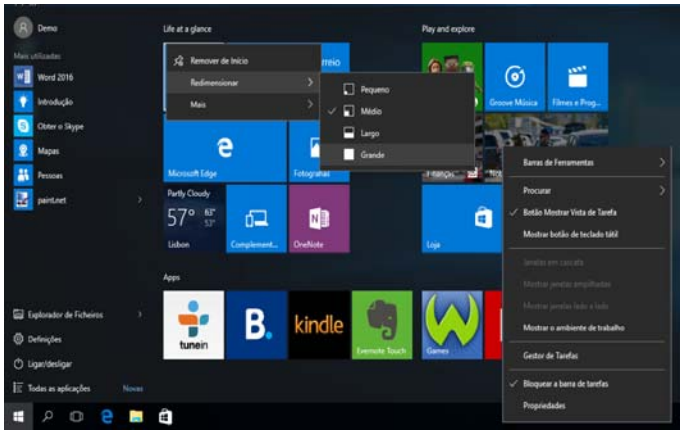
Uma das funcionalidades do **Windows 10** que recebeu melhorias nesta Build foi o **Menu Iniciar**. Apesar das mudanças serem subtis à primeira vista, passa agora a ser possível **adicionar** mais uma **coluna** de **mosaicos dinâmicos**, aumentando assim de **três** para **quatro** a quantidade de mosaicos **visíveis**. Para além disso, a Microsoft alterou o **número** de **itens** que podemos ter disponíveis no **Menu Iniciar** de **512** para **2048.**

Se quisermos então ter mais uma coluna disponível no **Menu Iniciar**, podemos aceder a **Definições > Personalização > Inicio** e **ativar** a opção **"Mostrar mais mosaicos"**.

Menu Iniciar

Menus de Contexto

Os **menus de contexto** nesta Build melhoraram significativamente quando comparamos com os existentes na **versão RTM**. Apesar de ainda existirem algumas diferenças entre os variados menus de contexto do Windows 10, as melhorias implementadas agora nesta versão tornam estes menus mais consistentes entre si. Para além das melhorias no **UI**, o menu de contexto possui agora **submenus** - no caso do **Menu Iniciar** - que dão um acesso rápido às opções que permitem definir a **dimensão** dos **Tiles**, **Rever** e **Votar** em **Aplicações** ou aceder a opções de **Partilha**.

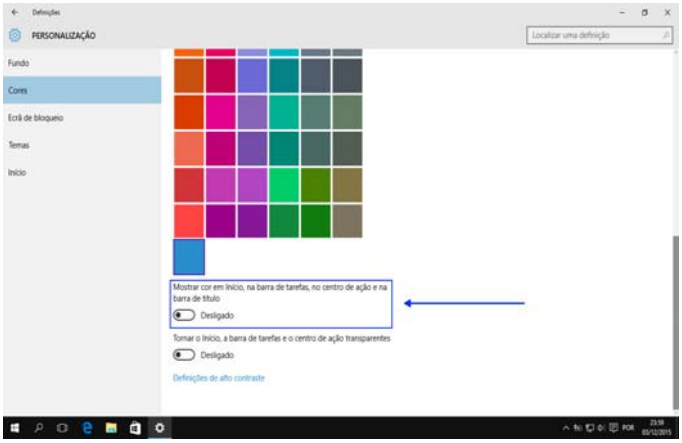


Menus de contexto

Cores na Barra de Titulo das Janelas

Na versão RTM do Windows 10, apenas era possível definir **cores** para **o Menu Iniciar**, **barra de tarefas** e **Centro de Ações**. Na Build 10586, é agora possível expandir o esquema de cores do Windows também às **barras de títulos**. Esta é com certeza uma das novidades que agrada à maioria dos utilizadores do Windows 10.

Para ativar esta opção, podemos aceder a **Definições > Personalização > Cores** e **ativar** a opção **"Mostrar cor em Inicio, na barra de tarefas, no centro de ação e na barra de títulos".**

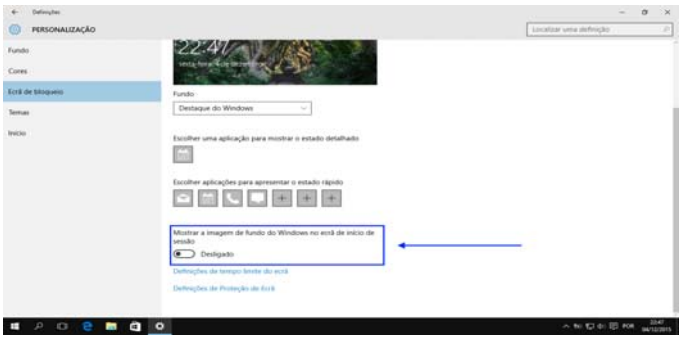


Esquema de cores

Imagem de Fundo do ecrã de login

A impossibilidade de inativar a **imagem de fundo** no **ecrã de login** do Windows 10, recebeu um grande número de feedback por parte dos **Windows Insiders** e que se traduziu numa nova opção de personalização. Na **Build 10586** passa então a possível **inativar** a **imagem de fundo** do ecrã de login.

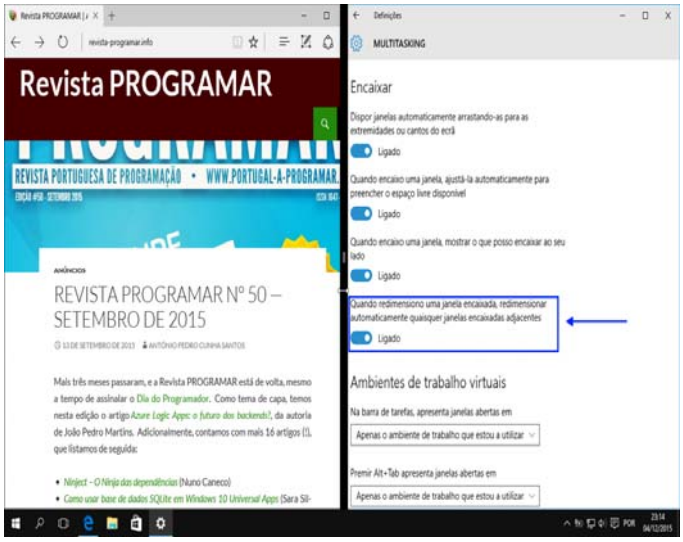
Esta nova opção está acessível em **Definições > Personalização > Ecrã de bloqueio > Mostrar imagem de fundo do Windows no ecrã de inicio de sessão.**



Personalização ecrã de login

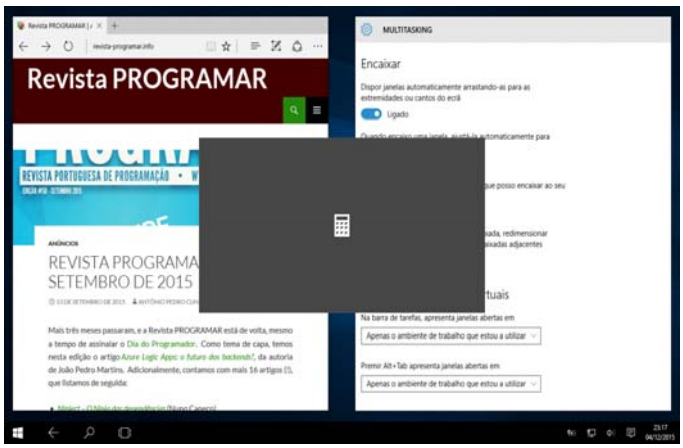
Multitasking, Snap e Tablet Mode

As opções de **Multitasking** no **Windows 10**, permitem configurar o comportamento das janelas (em **Snap**) no desktop e também nos desktops virtuais. Na atualização de novembro, foram melhoradas algumas das funcionalidades existentes na versão RTM e incluídas mais algumas opções, como por exemplo a opção **"Quando redimensiono uma janela encaixada, redimensionar automaticamente quaisquer janelas encaixadas adjacentes"**. Em termos práticos, quando redimensionamos uma janela que esteja no modo **Snap**, a outra janela é redimensionada de forma automática para que uma não se sobreponha à outra. Aqui temos claramente uma grande melhoria, considerando que esta opção estava apenas disponível no **modo Tablet**.



Multitasking

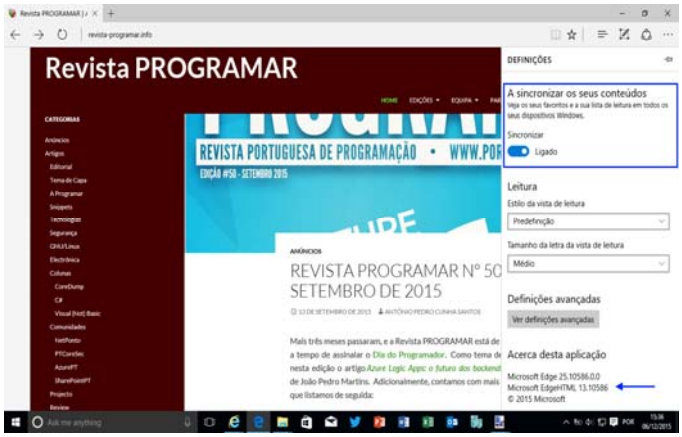
Outra nas novidades do Windows 10 - o **modo Tablet** - recebeu melhorias nesta versão que tornam a sua utilização mais simples e fluida. Agora quando usamos a **Vista de Tarefas** no modo Tablet, é possível **mover** Apps da esquerda para a direita ou **substituir** uma App encaixada anteriormente por uma nova App. Uma outra novidade do modo Tablet, é a possibilidade de **arrastar** qualquer App para o fundo do ecrã para que a mesma seja **fechada**. Se era um utilizador do **Windows 8.1**, de certeza que as funcionalidades referidas anteriormente lhe soam a algo familiar.



Modo Tablet

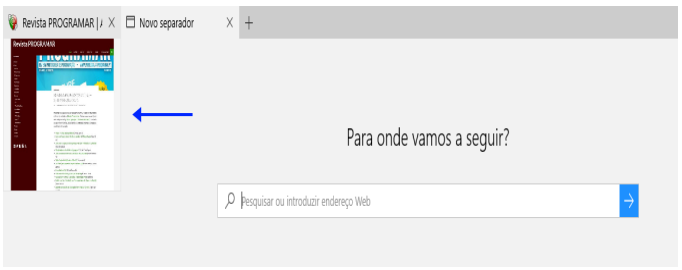
Microsoft Edge

O novo browser do Windows 10 - o **Microsoft Edge** - recebeu com a atualização de novembro a sua primeira grande atualização de plataforma, passando para a versão **EdgeHTML 13.10586**. Além desta atualização, o Edge recebeu um conjunto de funcionalidades que não foram incluídas na versão RTM do Windows 10, tais como a **sincronização** de **Favoritos**, **Listas de Leitura** e **Definições** entre PCs com Windows 10.



Sincronização Favoritos/Lista de Leitura

Também nesta versão, podemos **pré-visualizar** os sites abertos nos diversos **separadores**, bastando para isso passar o rato por cima de qualquer um destes.



Pré-visualização separadores Edge

A partir de agora, passa a ser possível também utilizar o Edge para fazer **casting** de conteúdos como **vídeo**, **áudio** e **imagens** para qualquer dispositivo externo que possua **Miracast** ou **DLNA**, como **uma Xbox One** por exemplo, excetuando obviamente conteúdos protegidos.

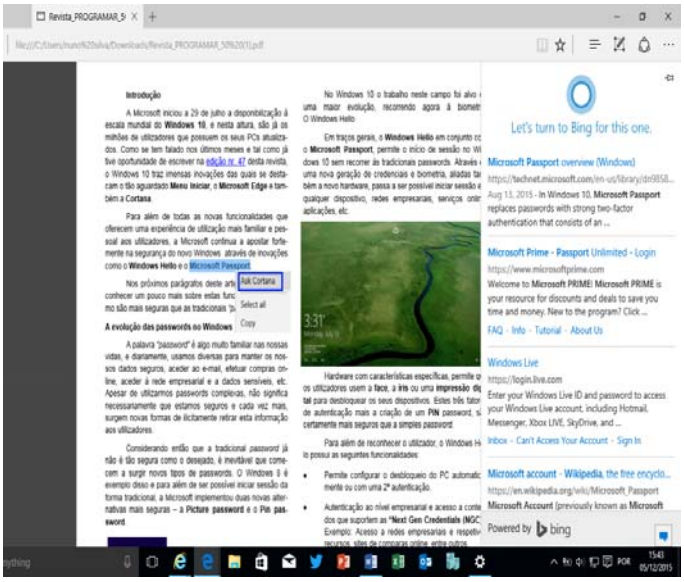
Casting no Microsoft Edge

Cortana

A Microsoft continua a apostar fortemente na **Cortana**, capacitando-a através da implementação de novas funcionalidades e melhorando algumas das existentes. Com a atualização de novembro, a Cortana passa a estar disponível no **Japão, Austrália, Canadá** e **Índia** (em inglês), com funcionalidades e experiências específicas para estes mercados. Considerando a dimensão do mercado português, continuaremos ansiosamente a aguardar a disponibilização da Cortana na língua de Camões.

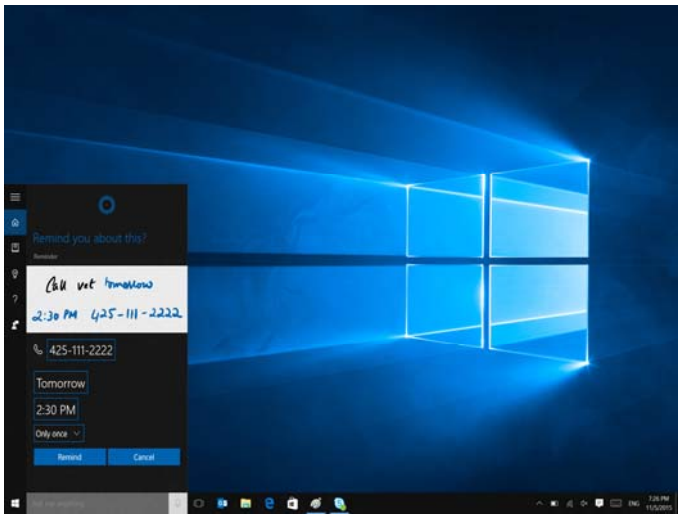
Outra novidade incluída nesta versão e que surge das inúmeras solicitações dos **Windows Insiders**, é a possibilidade de utilizar a Cortana também com **contas locais**, ou seja, cai a obrigatoriedade de configurar uma **Microsoft Account** para **iniciar sessão** no PC, exclusivamente para utilizar recursos como a **Cortana**, embora se mantenha a obrigatoriedade de indicar a **Microsoft Account** durante o processo inicial de **setup** da Cortana. Esta possibilidade é muito bem-vinda, tendo em conta que o mesmo já era possível com outras **Apps Windows**, como a **Store** por exemplo.

A opção **"Ask Cortana"** existente no **Edge**, passou a estar disponível também nos documentos **PDF**. Para obtermos informações sobre determinada **palavra** ou **frase**, basta **selecionar** esse conteúdo e com o **botão direito** do **rato**, **clicar** em **"Ask Cortana"**. Imediatamente a Cortana vai analisar o pedido e apresentar um conjunto de resultados, como ilustra a imagem que se segue.



Ask Cortana - PDF

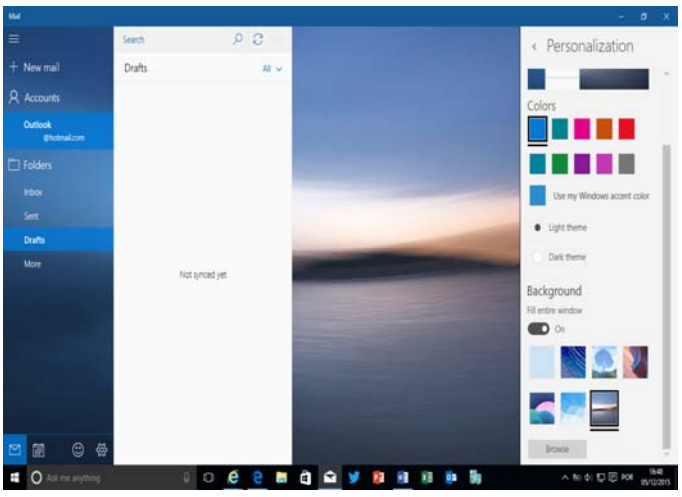
A Cortana recebeu melhorias na gestão dos nossos lembretes, passando agora a **incluir** nos **alertas** informações sobre **reservas** de **bilhetes de cinema**, de **viagem** e **encomendas** a partir das informações sobre estas e que recebemos normalmente no nosso **e-mail**. Outra novidade da Cortana e que está ligada também aos lembretes, é o **reconhecimento** de apontamentos feitos **manualmente** pelos utilizadores. Desta forma, informações como **datas/horas**, **números de telefone**, **moradas** e **e-mails** podem ser automaticamente utilizados nos **lembretes**.



Reconhecimento escrita Cortana

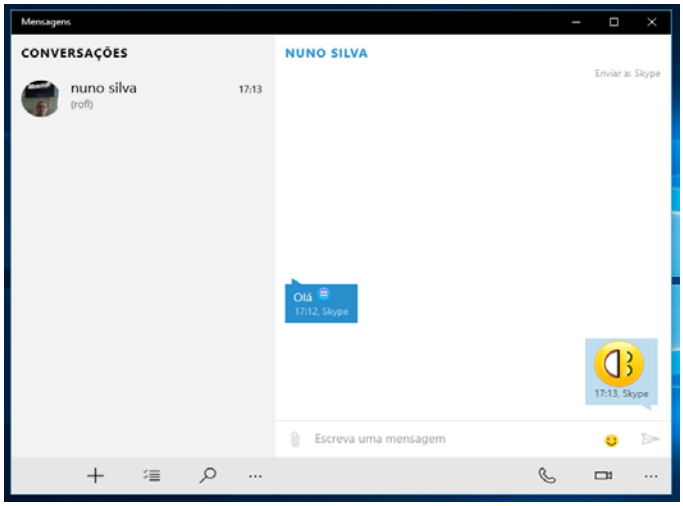
Windows Apps

Como tem sido habitual, também as Apps nativas do Windows 10 são atualizadas com novas funcionalidades e correções que melhoram significativamente a sua performance. Nesta Build foram atualizadas as Apps **Correio e Calendário**, **Fotografias**, **Mapas**, **Groove** e também **Xbox.** No Windows Blog, poderá encontrar mais detalhes sobre as novidades incluídas em cada uma destas Apps. Adicionalmente, a App **Feedback** passa a estar disponível para todos os utilizadores do **Windows 10**, mesmo que não estejam inscritos no **Windows Insider Program.**



App Correio

Foram também incluídas três novas **Universal Apps**: **Mensagens**, **Telefone** e **Skype Vídeo**. Através da integração de um conjunto de funcionalidades do **Skype**, é possível fazer chamadas de voz e vídeo ou enviar mensagens de forma rápida para outros utilizadores Skype, a partir do **Menu Iniciar** ou da **barra de tarefas**.



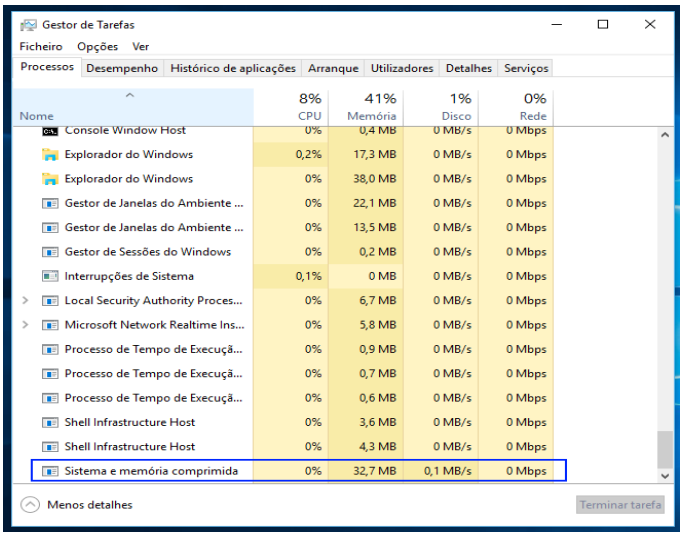
App Mensagens

Ainda relacionado com as **Apps**, a Microsoft voltou a disponibilizar a opção que permite a **instalação** de Apps em **dispositivos externos** como **cartões SD**. Esta opção foi apresentada inicialmente em algumas **Technical Previews**, mas não foi incluída na **versão RTM** do Windows 10.

Esta opção pode ser configurada em **Definições > Sistema > Armazenamento** > **"As novas aplicações serão guardadas em".**

Gestão da Memória

Em regra geral, o Windows mantém itens em memória para melhorar a performance, mas quando a memória começa a esgotar, começa a usar cache em disco para armazenar estes itens. Nesta Build, a **gestão de memória** inclui um novo componente que se chama **"System and compressed memory"**. Este componente permite que o Windows faça a **compressão** de informação na **memória** antes que esta se esgote, e antes que esta seja passada para o **disco**.



Gestor de Tarefas - Sistema e memória comprimida

Para saber um pouco mais sobre a **compressão da memória** no Windows 10, poderá consultar o seguinte vídeo no **Channel 9**: Memory Compression in Windows 10 RTM.

Enterprise Features

Como referi na introdução deste artigo, a **atualização de novembro** não traz apenas novidades para o consumo. As empresas que usam **Windows 10**, passam a usufruir de um conjunto de serviços e funcionalidades que permitem uma atualização continua dos seus equipamentos, mantendo-os sempre funcionais e seguros. Estas melhorias ajudam as equipas de IT a gerir uma grande quantidade de dispositivos sejam eles propriedade da empresa ou dos seus utilizadores (**BYOD**), manter a segurança da informação empresarial isolada da informação pessoal e autorizar a utilização de Apps disponibilizadas pela empresa em dispositivos empresariais e vice-versa.

Vejamos então algumas dessas novidades:

Windows Update for Business e Windows Store for Business

Estes dois serviços **gratuitos** disponibilizados pela Microsoft, vão permitir às equipas de IT gerir a forma como são **distribuídos** os **Updates** nestes equipamentos, assegurando que os mesmos estão atualizados e que cumprem os requisitos de segurança impostos pela organização.

Através da **Store for Business**, as equipas de IT podem adquirir e distribuir Apps para os dispositivos com Windows 10 de forma prática, seja estas Apps da **Loja Windows** ou App desenvolvidas internamente pela

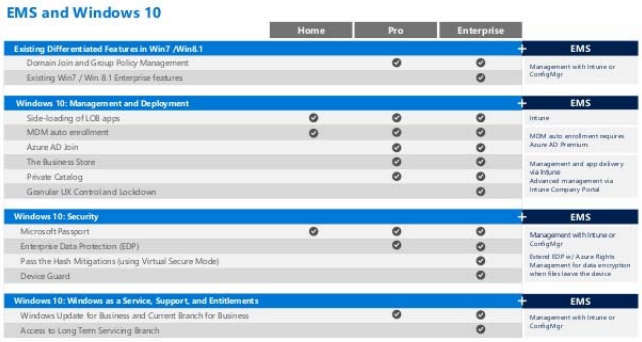
empresa **(LOB).**

Windows Store for Business

Mais detalhes sobre estes serviços, podem ser consultadas nos seguintes **links**: Windows Update for Business | Windows Store for Business.

Mobile Device Management e Azure Active Directory Join

Através do **Enterprise Mobility Management**, as equipas de IT vão poder gerir de forma eficiente os cenários de **BYOD** que podem incluir os mais variados dispositivos da



família **Windows**, como **PCs**, **Smartphones**, **Tablets** e também **IoT**.

Enterprise Mobility Management

Com a introdução do **Azure Active Directory Join**, qualquer dispositivo estará pronto a usar dentro da organização em poucos minutos. Através de um **login** único, os utilizadores podem fazer iniciar sessão num dispositivo e ter as suas **definições** do **Windows** e demais informações, migradas rapidamente e em segurança.

Mais detalhes sobre estas funcionalidades, podem ser consultadas nos seguintes links: Enterprise management for Windows 10 devices | Azure Active Directory Join

Segurança

A propagação de **malware** que tem como objetivo a subtração de **informação confidencial** é cada vez mais frequente. A Microsoft desenvolveu o **Windows 10** para proteger os seus utilizadores deste tipo de ameaças, através de funcionalidades como o **Windows Hello, o Windows Defender, o Device Guard e o Credential Guard**, agora melhoradas com a atualização de novembro.

Windows 10 Security

Mais detalhes sobre funcionalidades como o Device Guard, Credential Guard, Enterprise Data Protection etc.,

podem ser consultados no seguinte link: Microsoft Technet – Windows 10 Enterprise Features.

> ***Uma das funcionalidades do Windows 10 que recebeu melhorias nesta Build foi o Menu Iniciar. Apesar das mudanças serem subtis à primeira vista (...)***

Conclusão

Em conclusão, a **Build 10586** chega aos utilizadores estável e com melhorias significativas em termos de performance e experiência de utilização. O processo de instalação é em tudo igual à **versão RTM**, pelo que não deverá trazer problemas à maioria dos utilizadores. De referir apenas que existem outras novidades que merecem ser exploradas e que não foram abordadas neste artigo. Caso ainda não tenha atualizado para a Build 10586, não deixe então de explorar estas e outras novidades da versão.

> ***No Windows Blog, poderá encontrar mais detalhes sobre as novidades incluídas em cada uma destas Apps (...)***

AUTOR

Escrito por Nuno Silva

Microsoft MVP Windows Experience | Microsoft Technical Beta Tester

ECMASCRIPT 2015: A NOVA VERSÃO DO JAVASCRIPT

Este artigo não pretende mais do que apenas introduzir e clarificar alguns conceitos. Tecnicamente há muito para falarmos e, quem sabe, não estará na hora de programarmos juntos (em futuras edições da "nossa" revista)… Entretanto, se o leitor é programador de JavaScript, chegou a altura de dedicar algum do seu tempo à nova versão do ECMAScript, caso ainda não o tenha feito.

Afinal o que é o ECMAScript 2015?

A linguagem de programação JavaScript segue um padrão definido pela ECMA sob o nome de ECMAScript. Por outra perspetiva, podemos afirmar que o ECMAScript é o nome "apropriado", ou formal, para a linguagem usualmente referida como JavaScript. Objetivamente, sem grandes contextualizações históricas, o ECMAScript 2015 é a nova versão da especificação da linguagem JavaScript. É, muitas das vezes, também referido como ECMAScript 6, abreviadamente ES6, ou "Harmony".

As principais características da linguagem JavaScript são definidos pela norma ECMA-262 que está na 6ª versão e data, em versão final, do mês de junho. Esta última versão representa a maior mudança para o JavaScript, observada desde a criação do mesmo. Neste contexto, confesso-vos que alterei este artigo porque, quando iniciei a redação do mesmo (15 de junho), a referida norma, na sua 6ª versão, existia apenas em versão draft, com a data de 13 de abril último, sendo que no dia 17 de junho sou agradavelmente surpreendido com o lançamento da versão final do ECMAScript 2015 (nome dado à norma ECMA-262 na sua 6ª Edição).

As versões anteriores do ECMAScript são as 1,2,3 e 5. A omissão do 4 é factual e não um erro. O ES5 teve o seu lançamento em 2009, sendo que o ES6 (ou ES2015) é trabalhado desde então.

Consultar versões draft do ECMAScript: http://wiki.ecmascript.org/doku.php?id=harmony:specification_drafts

Versão final do ECMAScript 2015: http://www.ecma-international.org/ecma-262/6.0/index.html

Com tantas designações para, aparentemente, a mesma coisa, deixo abaixo uma tabela com o intuito de dissipar qualquer dúvida que ainda permaneça no leitor, focando-me no essencial.

ECMAScript 2015	Especificação em que o JavaScript se baseia
JavaScript	Implementação da especificação EC-
ES6	Versão do ECMAScript (*old style*)
ES2015	Versão do ECMAScript (*new style*)

O JavaScript

Já com 20 anos de existência, quantos de nós já utilizaram o JavaScript para a definição de algumas variáveis, para cálculos simples, para controlo do UI (User Interface) ou para a validação de formulários? Muitos, senão todos… Mas este tipo de utilização da linguagem era típica dos seus inícios e efetivamente restringia-se a isso e pouco mais. A evolução, até aos dias de hoje foi tremenda e, atualmente, o JavaScript passou o seu foco para a construção de aplicações robustas, quer do lado do cliente, quer do lado do servidor. Por via do dinamismo de uma comunidade de verdadeiros devotos à linguagem e devido à capacidade de ser executado em plataformas distintas, o seu crescimento e consequente notoriedade são evidentes. Há, ainda, quem considere que o JavaScript não dispõem das caraterísticas mais adequadas à boa produtividade de um programador e não facilita a produção de código de fácil manutenção. O surgimento do ES2015 também tem o intuito de colmatar algumas dessas falhas, colocando a linguagem a um outro nível.

Há duas vantagens factuais na utilização do JavaScript:

Ubiquidade:

podemos encontrá-lo em qualquer dispositivo, desde os *smartphones* até qualquer computador atual. Qualquer programador que já tenha desenvolvido para a web, se não o utilizou, pelo menos pensou na sua utilização. Embora, neste particular, nem consiga sequer conceber o desenvolvimento para a web sem o recurso ao JavaScript.

Poder:

A linguagem revela-se extremamente capaz especialmente em dar resposta à, de cada vez maior, necessidade aplicacional de comunicações assíncronas (caso estas sejam necessárias).

Nós, programadores, já percebemos que o *browser* é a forma que temos disponível para, de forma eficaz, criar interfaces de utilizador para a partilha de dados e que de cada vez mais a nossa atividade tende a ser para a construção de aplicações web, em detrimento das GUI clássicas. O JavaScript é peça chave desta mudança de paradigma. Em reforço disto, um browser - e, portanto, o JavaScript - pode ser executado numa vastíssima gama de dispositivos, desde telemóveis com CPUs eventualmente com pouca capacidade de processamento, até computadores poderosos com enorme capacidade de processamento.

Curiosamente, no mesmo mês do lançamento do ES2015, a palavra JavaScript passou a fazer parte do dicio-

nário. Por agora ainda não faz parte do português, foi um dicionário de língua inglesa de referência, o Oxford English Dictionary, que o incluiu na listagem das novas palavras.

Atualizações ao referido dicionário: http://public.oed.com/the-oed-today/recent-updates-to-the-oed/june-2015-update/new-words-list-june-2015/

Podemos utilizar o ES2015 desde já?

Com entusiasmo diria, no imediato: "claro que sim".

Muitas das especificidades do ES2015 já são disponibilizadas pelos browsers Firefox e Chrome (consultar https://kangax.github.io/compat-table/es6/). No caso deste último poderá ser necessária a ativação da opção **Ativar JavaScript Experimental (***Enable Experimental JavaScript)* acessível nas configurações através do acesso a chrome://flags (na barra de endereço do browser).

Numa abordagem muito pouco técnica, mas elucidativa, o ES2015 atualiza o JavaScript. Neste contexto, imagine-se o cenário em que temos uma aplicação estável com funcionalidades suportadas pelo JavaScript (exclusivamente ES5, no caso). Na realidade temos disponível, para programar sobre um browser atual e atualizado, tudo o que sabemos da especificação da linguagem definida pelo ES5, somada a melhorias desta última, mais as novidades trazidas pelo ES6. Então como proceder? Recomendaria a programação em ES2015 testando, sempre, o código em diferentes browsers, de resto como é prática comum nos programadores de aplicações web.

Em suma, o desafio traduz-se em utilizar o ES2015 gradualmente nos nossos projetos, aproveitando as melhorias/atualizações do que já existia no ES5 e tirando partido das novidades trazidas pela nova versão. E não esqueçam que, hoje, as nossas aplicações em JavaScript são executadas em browsers que já suportam o ES2015.

Indo um pouco mais longe, existem *transpilers* (*source-to-source compilers*) que traduzem o código integralmente escrito em ES2015 para a versão anterior, a ES5, integralmente suportada pelos browsers atuais. Neste contexto refiro o Babel (disponível em https://babeljs.io/) que, à data, é considerado o melhor *transpiler* de JS pela comunidade. Em suma, com o Babel corretamente instalado, podemos produzir aplicações integralmente escritas em ES2015 sem ter qualquer preocupação de compatibilidade com os *browsers*.

Como diz o slogan utilizado no site do Babel: *Use next generation JavaScript, today*. Subscrevo.

Conclusão

O ECMAScript2015 mudará (já muda) a forma como se programa em JavaScript.

Em contexto de produção, neste momento, deparamo-nos com um problema que devemos ter em consideração: o suporte nas várias plataformas disponíveis. Com efeito, o ES2015 ainda não é integralmente suportado pelos browsers, embora se assista a constantes e rápidas evoluções nesse sentido.

Na minha opinião, não faz grande sentido intervir numa aplicação terminada e já em produção e migrá-la para a nova versão do JavaScript. O mesmo se aplica a uma aplicação num estágio de desenvolvimento já avançado. No entanto, já faz todo o sentido a utilização do ES2015, no momento em que é iniciado um novo desenvolvimento aplicacional. Sempre de considerar é o suporte (ou a falta dele) por parte dos browsers.

Concluíndo e clarificando, enquanto programadores de JavaScript, não temos a necessidade de aprender um JavaScript "novo". Podemos continuar a programar utilizando todo o nosso conhecimento e sintaxe aprendida até então, tirando partido das melhorias, do que já existia, agora introduzidas pelo ECMAScript 2015 e, sempre que necessário, tirar partido do que o referido *standard* nos traz de novo.

Como referido na introdução gostaria de voltar com, pelo menos, a apresentação do que, na prática, o ES2015 nos proporciona. A ver vamos… Se for caso disso um "até já" e foi um prazer escrever para vocês.

AUTOR

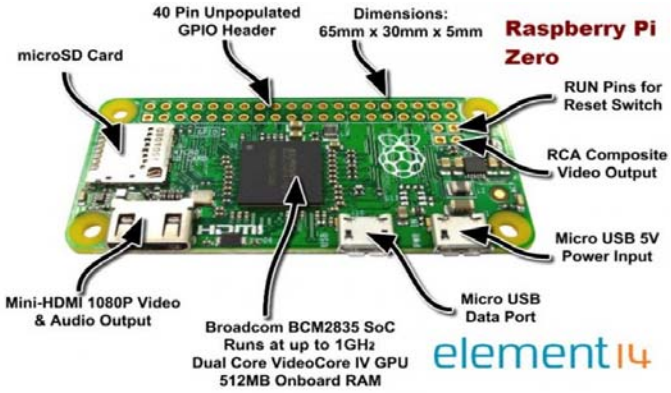
Escrito por José Viana

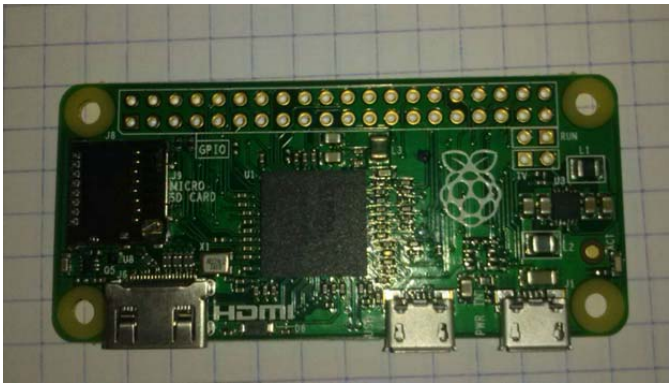
exerce profissionalmente a atividade de programador desde há, aproximadamente, 15 anos. Docente da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Viana do Castelo, na Área Científica de Informática e Multimédia.

Licenciado em Engenharia da Computação Gráfica e Multimédia.

RASPBERRY PI ZERO – O NOVO MEMBRO DA FAMÍLIA!

Para os amantes da família Raspberry Pi, o Natal chegou mais cedo. O final de Novembro trouxe um novo membro à família Pi. O Zero. Dono de 1Ghz, single-core CPU e de 512MB RAM, as espectativas geradas em torno deste lançamento eram altas. O Zero não desiludiu quem o esperava impacientemente.





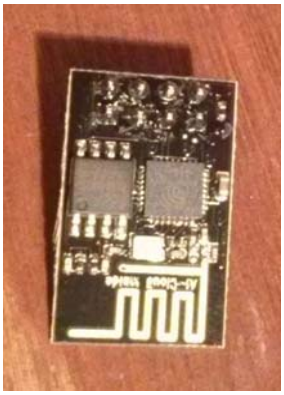
Como o próprio nome indica, quase que voltámos ao inicio. À simplicidade de um circuito. De uma ideia. Mais compacto, o novo Pi Zero é ligeiramente mais pequeno que o Raspberry Pi A+. Dimensões oficiais? 6.5cmx3.52cm. É micro. Mas só no tamanho. As potencialidades são intensas (e imensas), bem à forma como a fundação Raspberry nos habituou.

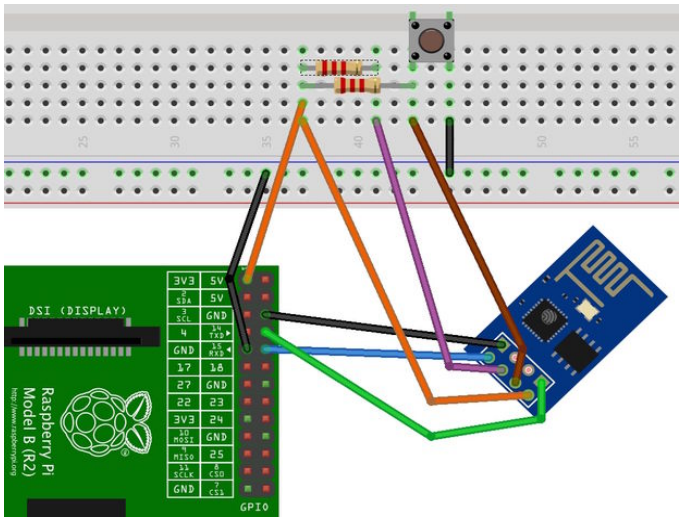
Como o Zero não é só mais um membro desta família, sendo que aflora toda a vertente mais técnica dos utilizadores, vamos a comentários mais práticos… a porta USB passou a ter uma ficha micro USB. A porta HDMI passou a ser mini-HDMI. Caso seja necessário, podem (e devem) ser usados adaptadores externos (por exemplo para ligar ao monitor de casa).

Deixamos de ter os "pinos" de origem, nesta versão os headers deixam de estar soldados e são considerados um extra que o utilizador tem que tratar à posteriori, caso decida ligar por exemplo um flat-cable e/ou para que haja interacção entre o Pi Zero e outros circuitos. Outra vertente que deixamos de ter directamente é a saída S-VIDEO, também esta terá que ser soldada pelo utilizador.

Contudo à primeira olhadela o que salta mais depressa à vista é o facto do Raspberry Pi Zero não ter interface de rede.

Os mais distraídos podem ficar descansados pois a pequena placa pode ter acesso à internet. Basta para isso utilizar uma interface USB-Wifi. Para os mais corajosos e entusiastas da electrónica, podem ser usados os pinos GPIO14, GPI015, +5V, GND e duas resistências pull-up, para ligar o módulo WIFI UART, baseado em ESP8266 (expansão GPIO do raspberry para ligar um modulo ESP8266-01, como mostra a imagem seguinte, por exemplo).





Modulo ESP8266-01

Diagrama de ligação ao modulo

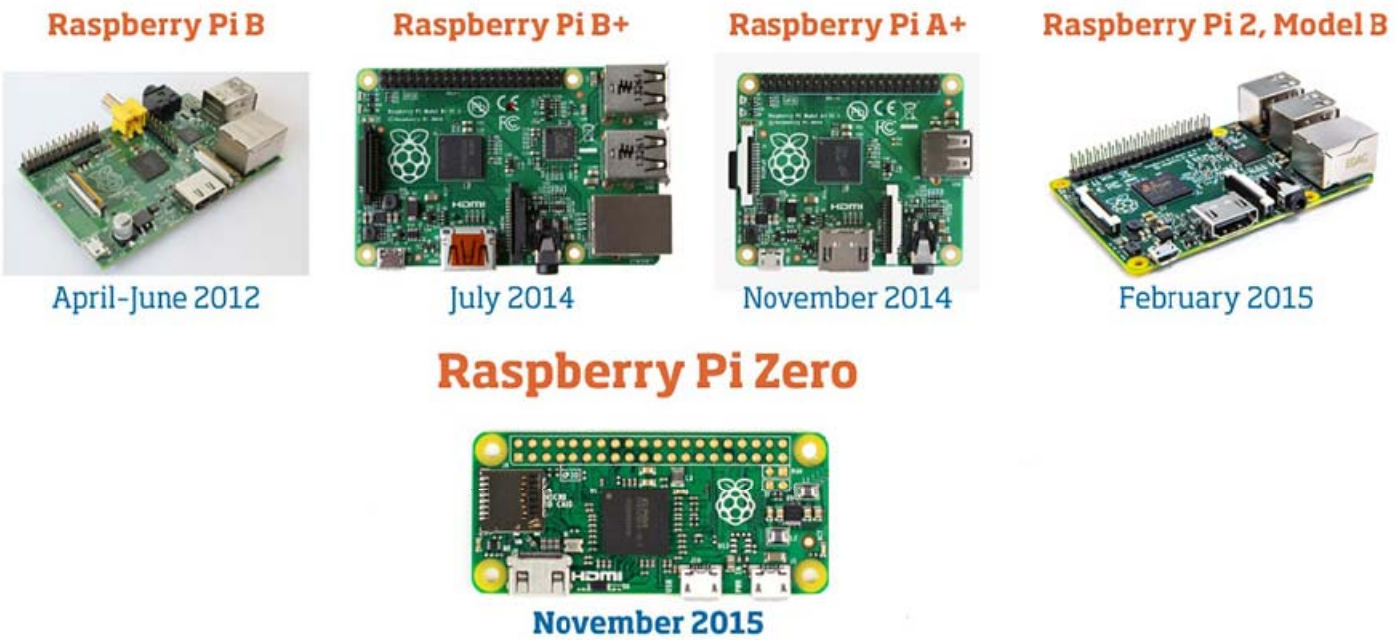
RASPBERRY PI ZERO – O NOVO MEMBRO DA FAMÍLIA!

Apesar de teoricamente esta nova versão prometer cerca de 40% maior rapidez de resposta, na prática isso não se vislumbra propriamente. Contudo o Zero cumpre o que promete, e é o elemento mais pequeno da família, o que consome menos energia e também o mais económico aquando da aquisição.

> ***Como o próprio nome indica, quase que voltámos ao inicio. À simplicidade de um circuito.***

Escusado será dizer que à semelhança do que aconteceu anteriormente, também esta versão do Raspberry esgotou nos primeiros dias, a primeira edição de 200.000 exemplares, uma vez que apesar de parecer "incompleta", é a versão da família Pi que permite ao utilizador uma maior decisão e interacção/construção.

Em suma, apesar de toda a família Raspberry Pi ser destinada ao público em geral, uma vez que o mote da fundação é ensinar de forma simples e económica todos os interessados em aprender e dar os primeiros passos na programação, o Raspberry Pi Zero afasta-se um pouco do público em geral sendo que é quase um pequeno competidor do arduino, sendo mais atractivo às pessoas que estejam a traba-



AUTOR

Escrito por Rita Peres

Natural de Castelo Branco, licenciou-se em Engenharia Informática pela Universidade da Beira Interior. Membro do P@P desde Janeiro de 2010.

BETA-I HACKATHON LISBON HAS BEEN HACKED

Nos passados dias 24 e 25 de Outubro, Lisboa foi palco de uma das maiores (e melhores) iniciativas dos amantes da programação. Novos e menos novos, juniores, seniores ou apenas aspirantes a designers/programadores juntaram-se nas instalações da Beta-i para quebrar barreiras.

Como não podia deixar de ser, a Programar esteve nesta iniciativa. No sábado de manhã, vários foram os audazes que se dirigiram ao local com um sorriso no rosto, registos feitos, foram abertas as hostes. Apresentadas ideias, formadas as equipas e dai ao "work & code" foi um rápido passo. A meio da manhã, já se debatiam ideias e algoritmos sendo que os primeiros esboços foram rapidamente surgindo.

O espaço era agradável, com espaços para programar e desanuviar enquanto as ideias vinham preenchendo as mentes dos programadores.

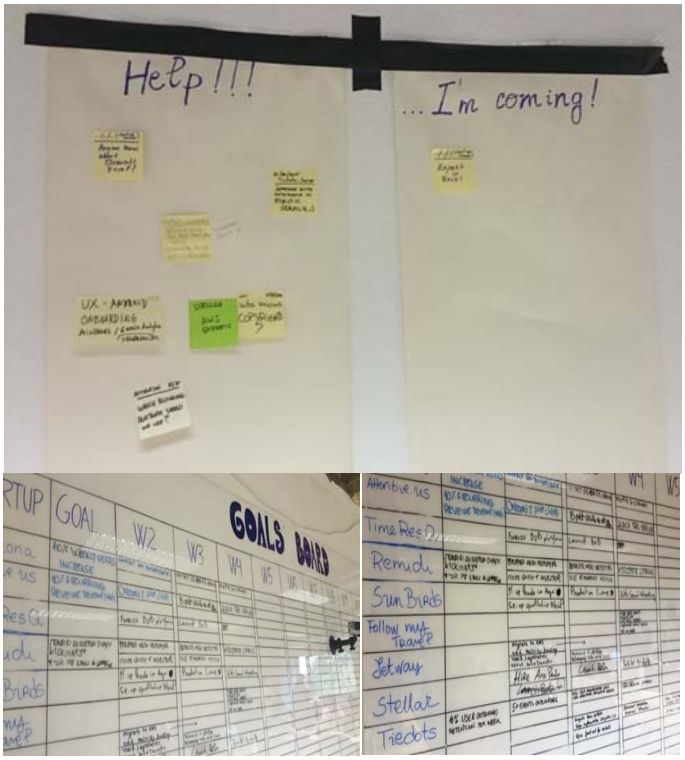
Trabalhos divididos em quatro grandes grupos: "The Rebels", "The Explorers", "The Makers" e "The Revolutionaries", os projectos foram crescendo.

Num ambiente em que a boa disposição reinou, foram vários os post-its deixados nos quadros e recantos espalhados pela sala, o "Goals Board" ajudava a manter presentes os objectivos e metas a atingir. As ideias transformaram-se em projectos, os projectos ganharam novos forks e tudo correu em beleza.

Não menos importante e como alguém tinha que cuidar dos participantes, a organização não teve mãos a medir, sendo que ao longo dos dias, não deixaram ninguém "à fome", pelo contrário, foi servido um almoço bastante saudável, e a zona do lanche e do café não deixou de ter sempre algo apetitoso. Ora… comida, café e código… qual o programador que não se sentiria em casa?

No Code

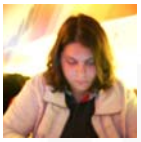
BETA-I HACKATHON LISBON HAS BEEN HACKED



> ***"Apresentadas ideias, formadas as equipas e dai ao "work & code" foi um rápido passo (...)"***

Foto by: Filipa Peres

AUTOR



Escrito por Rita Peres

Natural de Castelo Branco, licenciou-se em Engenharia Informática pela Universidade da Beira Interior. Membro do P@P desde Janeiro de 2010.

XAMARIN 4.0 – TUDO O QUE PRECISA PARA CRIAR APLICAÇÕES MÓVEIS

Introdução

No primeiro semestre de 2014 a Xamarin lançou o Xamarin 3.0, que veio trazer às equipas de desenvolvimento de aplicações móveis muitas novidades, nomeadamente a framework Xamarin Forms que veio revolucionar a forma de desenvolvimento até então e para além disso possibilitou a criação de estratégias de reutilização de código, que permite diminuir o esforço de desenvolvimento entre as várias plataformas e claro está reduzir o tempo de manutenção.

Com isto, nos últimos meses muita tinta correu à volta das soluções oferecidas pela Xamarin, e com a publicação do Xamarin Test Cloud, Xamarin Insights e Xamarin Android Player, a Xamarin continuou a somar pontos.

Recentemente, em Novembro 20015, a Xamarin volta à carga com mais anúncios e apresenta o Xamarin 4.0, que apresenta muitas novidades que para quem está a fazer desenvolvimento de aplicações móveis deixa qualquer um soberbo, uma vez que a qualidade e leque de funcionalidades continua a crescer de forma exponencial, e consequentemente contribui para o bom desenvolvimento, e consequentemente boa qualidade e desempenho das aplicações.

Funcionalidades

O Xamarin 4.0 apresenta-se com uma nova versão da plataforma da Xamarin e da Xamarin Test Cloud, e torna público o Xamarin Insights, deixando este último de ser versão beta, como era até aqui.

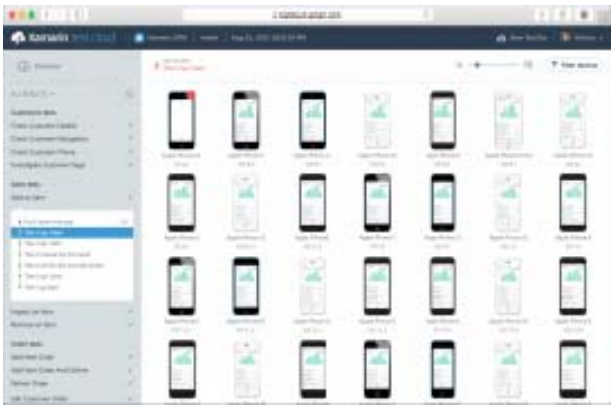
As soluções da Xamarin vão fornecer ferramentas e APIs para facilitar o desenvolvimento, teste e monitorização, e claro está as estratégias de partilhar/reutilização de código permite que o desenvolvimento das aplicações para as várias plataformas seja mais rápido.

Ao nível da framework Xamarin Forms, que se apresenta como Xamarin Forms 2.0, temos as principais novidades:

- Melhorias de performance e de qualidade de API;
- Suporte para pré compilação de ecrãs desenhados em XAML facilitando o carregamento da aplicação;
- Suporte para Universal Windows Platform apps (ainda preview)
- Suporte para iOS9 e Android Material Design
- Novas “gestures”: pinch e pull-to-refresh

Para além destas novidades, a Xamarin apresenta melhorias no desenvolvimento de aplicações iOS usando Xamarin Plugin para Visual Studio, realizou atualizações na versão do Mono/.Net de forma a manter compatibilidades e aumentar a performance das suas soluções e melhorou a experiência ao nível de modo design para iOS (load and save XIB files) e Android (Android Material Design).

Ao nível do Xamarin Test Cloud, a Xamarin possibilita que as aplicações sejam testadas em +2k dispositivos reais de Android e iOS, o que sem dúvida é uma mais-valia para muitas equipas de desenvolvimento que muitas vezes só tem 2 ou 3 dispositivos. E é apresentado o Xamarin Test Recorder, aplicação “preview” disponível para Mac, que permite gravar todas as interações realizadas pelas aplicações iOS e Android e desta forma é possível criar scripts que depois podem correr no Xamarin Test Cloud ou importá-los para o Xamarin Studio/Visual Studio. Estes scripts baseiam-se em testes unitários que tiram partido da framework UITest e podem ser executados a partir de integração continua. Esta framework, Xamarin.UITest C#, é mais uma das novidades com o lançamento da sua primeira release e é completamente focada na criação de testes, é gratuita e sem limitações de utilização em simuladores ou devices. No entanto a combinação perfeita é o uso de Xamarin.UITest C# em conjunto com o Xamarin Test Cloud.



Ao nível do Xamarin Insights, a Xamarin orgulha-se na monitorização em tempo real e disponibiliza relatórios gratuitos para todos os utilizadores, no entanto com algumas limitações. Claro está fornece soluções mais avançadas para cenários que o exijam, o que é o caso das aplicações enterpri-

se.

Para terminar, a Xamarin aproveitou este lançamento para estimar os seus clientes e ofereceu a todos os utilizadores ativos:

- Relatórios de erros do Xamarin Insights, com retenção de dados até 30 dias;
- 60 minutos de Xamarin Test Cloud por mês
- 30 dias de acesso à Xamarin University

> ***(...) o Xamarin 4.0, que apresenta muitas novidades que para quem está a fazer desenvolvimento de aplicações móveis deixa qualquer um soberbo uma vez que a qualidade e leque de funcionalidades continua a crescer de forma exponencial(...)***

E para além disso a Xamarin apresenta novos planos sobre os seus produtos, planos estes mais direcionados a empresas.

Conclusão

Em conclusão, podemos concluir que a equipa da Xamarin continua a trabalhar arduamente para nos fornecer boas soluções de forma a facilitar o desenvolvimento de aplicações móveis e com o lançamento do Xamarin 4.0 nota-se a melhoria de qualidade e em muitos casos a voz do utilizador é tida em conta.

Referências

https://blog.xamarin.com/

http://developer.xamarin.com/

AUTOR

Escrito Por Sara Silva

Licenciada em Matemática pelo DMUC, e o seu foco de desenvolvimento está direccionado para a área Mobile. Atualmente desenvolve na área do Windows, Xamarin, Azure, e é Microsoft MVP Mentor. A Sara foi condecorada com vários prémios com especial destaque: Microsoft MVP, Xamarin MVP, Telerik Developer Especialista, C# Corner MVP, TechNet Wiki - Technical Guru. O trabalho que vai sendo desenvolvido pela Sara pode ser seguido através do seu blog www.saramgsilva.com e do twitter é @saramgsilva.

IMPRESSORAS 3D – ESCOLHE A TUA

Hoje em dia, as impressoras 3D ganham cada vez mais entusiastas e muitos são os interessados neste tipo de equipamento. Apesar da Impressão 3D não ser "algo novo" uma vez que a primeira impressão conhecida ocorreu em meados de 1984, por Chuck Hull, só há pouco tempo o público em geral olhou de forma mais atenta a este tema . A Impressão 3D também conhecida como prototipagem rápida, é uma forma de tecnologia de fabricação aditiva onde um modelo tridimensional é criado por sucessivas camadas de material.

Neste artigo, o objectivo será indicar um conjunto de impressoras 3D e respectivas análises dos seus prós e contras de forma que seja mais simples escolher o equipamento que melhor satisfaz as suas necessidades do leitor.

Impressoras para iniciantes

Da Vinci Jr. 1.0

Uma escolha acertada para quem procura uma impressora o mais económica possível e sem necessitar de qual quer tipo de montagem por parte do comprador, pois vem calibrada de fábrica, sendo simplesmente "plug and play". Sendo da família das impressoras XYZ, a Da Vinci Jr. 1.0 devido à sua relação preço\qualidade ganhou o prémio 2014 CES EDITOR´S CHOICE.

Esta pequena máquina tem a vantagem de ser completamente selada, evitando que durante a impressão ocorram movimentações ou acessibilidades indesejadas por parte de crianças ou objectos. O seu pequeno LCD apesar de básico, consegue manter um formato decente de forma a ser possível ler o seu conteúdo e relembrar que o seu cartão SD ainda está dentro do aparelho.

Na questão da sua qualidade, este equipamento oferece uma base que permite imprimir objectos até 200x200x200mm e a impressão é aceitável tendo em conta que o software é minimalista, não contendo muitas opções nas configurações da mesma para a impressão o que se torna um ponto favorável para quem inicia esta nova experiência. Quando o software inicia o "slicing", que é a designação dada ao processo onde o programa transforma o objecto tridimensional em "layers", parece que existe alguma dificuldade em processar objecto "menos perfeitos" e tende em demorar mais do que o desejado. À parte dos pequenos problemas de impressão, quero manter em mente a questão de preço\qualidade apesar deste fornecedor complicar uma acção tão simples como trocar o filamento sendo que o consumidor é impedido de comprar filamento mais acessível economicamente tendo que o comprar mesmo à empresa-mãe, este é um ponto que já dá que pensar duas vezes quando chega a hora de adquirir este modelo.

Em conclusão a esta análise, pode-se dizer que esta impressora se destina ao consumidor que pretende um equipamento o mais económico possível, que não seja muito exigente com o acabamento final da impressão e queira iniciar uma experiência neste mundo da impressão 3D.

Printrbot Play 1505

Uma impressora minimalista no visual e na qual esta versão tem a estrutura em metal, sendo tal como a Da Vinci Jr. 1.0, é uma impressora que vem montada de fabrica e não necessita de qualquer tipo de montagem. Ao contrário do que

aparenta, esta impressora não consta ser pesada pois grande parte é alumínio e a sua parte inferior é aberta permitindo assim um acesso simples e fácil aos motores, à electrónica e aos pés de borracha que impedem a mesma de deslizar sobre a mesa durante a impressão. Em comparação com os seus antecessores, este modelo contém um slot SD mais acessível e um apoio para o filamento no topo da estrutura.

Apesar de compacta, a estrutura contém tudo o que necessita de forma organizada, tal como o seu extrusor e as duas ventoinhas de arrefecimento, impedindo ainda que curiosos possam tocar em partes quentes ou rotativas deste aparelho. Contudo esta protecção torna-se uma obstrução caso seja necessário aceder ao seu interior, tendo que desmontar a estrutura exterior por completo.

As dimensões da Printrbot Play são reduzidas, tenho uma capacidade máxima de impressão de 100x100x130mm, realizando impressões rápidas sem ter de sacrificar significativamente a qualidade de impressão, tendo como única desvantagem a produção de ruído acima da média.

Dremel 3D Idea Builder

Chegamos a um nível a cima. Apesar de ter um preço um pouco mais elevado que as duas impressoras anteriores, a Idea Builder contém uma qualidade bastante aceitável. Sendo completamente fechada pela sua estrutura com uma porta frontal, contém um software simples de utilizar mas que infelizmente contém algumas falhas em aspectos fundamentais tais como a adição de material de suporte e exige também a utilização de filamentos vendidos pela empresa que na sua maioria são translúcidos e disponíveis em apenas 10 cores. Para além destes aspectos, esta impressora não tem uma base aquecida não sendo possível imprimir filamento de ABS.

Contudo, este modelo oferece um interface táctil e contém como ferramenta de apoio com vídeos que colocam o utilizador a realizar a sua primeira impressão em apenas 30min.

Rostock Max v2

Com um visual completamente diferente das impressoras analisadas até ao momento, a Rostock Max v2 oferece excelentes impressões sem ter de elevar o seu preço. Para além do visual, esta impressora não vem montada de fábrica mas o que pode ser um ponto a favor quando chega a hora de substituir algum dos componentes.

As suas dimensões não são propriamente simpáticas, contudo, para os leitores que nunca tiveram a oportunidade de ver este equipamento a trabalhar preparem-se pois é fascinante e invulgar. Ela imprime movimentando-se e deslizando nas três calhas para cima e para baixo. Cada calha contém um braço que está ligado ao extrusor que se movimenta consoante a movimentação desses mesmos braços. Com a colocação do extrusor nesse local, o peso é reduzido permitindo que surja uma movimentação rápida durante a impressão por parte dos braços. Devido a esta estrutura todos os componentes são de fácil acesso incluindo o interruptor e o cartão SD que está localizado lado a lado com o LCD.

Assim sendo, esta impressora será indicada para os principiantes mais destemidos, tendo que montar a mesma e calibrá-la, o que de início pode ser intimidativo mas uma excelente forma de entender de forma mais aprofundada como uma impressora 3D funciona fornecendo assim uma boa base de conhecimento para um futuro Upgrade.

As impressoras indicadas anteriormente são alguns dos modelos ideais para quem pretende iniciar esta vertente de prototipagem/modelação tridimensional sem ter de gastar valores elevados e montagens complicadas.

Por ultimo deixo aos leitores, uma outra impressora 3D, bastante conhecida no meio.

RepRap Pursa i3 Kit

Com um visual simplista e minimalista a Pursa i3, foi das primeiras impressoras 3D a ser amplamente conhecida, mais pelas comunidades de makers, uma vez que é relativamente simples de construir, bastante modular, e com um design extremamente minimalista. Sendo open-hardware deu origem a diversas impressoras, apesar de agora se encontrar disponível já montada e pronta a utilizar, bem como em kit de peças, todos os desenhos e diagramas estão disponíveis no github.

Não se pode dizer que seja uma impressora destinada aos mais iniciantes, a menos que já venha prémontada, sofre do pequeno mal de algumas peças serem feitas numa impressora 3D, o que lhes limita o ciclo de vida, no entanto é possível substituir essas peças, por peças metálicas produzidas numa máquina CNC , o que lhe aumenta a durabilidade.

O outro grande ponto forte das impressoras tipo Pursa i3, é a grande disponibilidade de peças compatíveis, o que permite uma grande variedade de upgrades, existindo de tudo desde cabeças extrusoras a quente, para várias dimensões e materiais, desde o ABS ao PLA, até polímeros metálicos, passando por extrusoras a frio para outros polímeros e materiais, até extrusoras de duas cabeças, painéis electrónicos com as mais variadas formas e funcionalidades, passando por toda panóplia de hardware intermutável.

Não se trata de uma impressora propriamente aconselhada aos mais principiantes, mas uma boa opção para os entusiastas, tendo uma grande variedade de modelos para os mais variados preços e sendo bastante modular, permitindo grandes melhoramentos pós aquisição.

Aos nossos leitores que ainda não estão totalmente familiarizados com as impressoras 3D quero deixar uma pequena ajuda, caso se entusiasmem e iniciem esta temática.

> ***A Impressão 3D também conhecida como prototipagem rápida, é uma forma de tecnologia e fabricação aditiva onde um modelo tridimensional é criado por sucessivas camadas de material.***

Polímeros

Um ponto a ter em atenção quando pensamos em impressoras 3D, são os polímeros que estas são capazes de utilizar. Por exemplo os polímeros de PLA apesar de biodegradáveis e bastante acessíveis, têm o mal de não oferecerem grande resistência térmica, ficando “moles”, quando a temperatura aumenta. São baratos, com uma pegada ecológica reduzida, mas não os ideais para todos os usos e para tarefas onde é exigida uma maior durabilidade, como por exemplo a produção de uma frame de um quadcopter. Por sua vez os polímeros de ABS, são claramente mais robustos, sendo um polímero muito usado na indústria das tecnologias, foi inclusive usado em armas, como o caso da famosa M16, utilizada pela primeira vez pelo exercito norte-americano na guerra do Vietnam, onde a primeira reacção ao uso de um polímero plástico numa arma fez com que fosse dito que “esta coisa, vai-se desfazer em combate”. Prova contrária deu a história provando que o ABS é um polímero tão fiável que neste momento é empregue até na indústria aeroespacial e de defesa. Existem diversas variantes do polímero de ABS, entre elas algumas resistentes ao fogo, até aos 270ºc com uma resistência tênsil de até 2440MPa e uma resistência de impacto de até 11HJ/m^2, passando por polímeros de carbono, nylon, e ABS transparente, (uma variante do polímero de ABS), bastante usado na indústria da moda, pela sua resistência ao impacto de aproximadamente 17Kj/m^2 e transparência, que torna este material apelativo para esta

actividade.

Claro que todos estes materiais, com as suas especificações variadas, muitas vezes requerem capacidades de impressão especificas, pelo que a finalidade de uma impressora e os materiais que ela deverá ser capaz de imprimir, permanece interligada.

> ***(...)a primeira impressão conhecida ocorreu em meados de 1984, por Chuck Hull (...)***

Conclusão

A impressão 3D, continua numa vertiginosa evolução, quer em termos de modelos de impressoras, cada vez mais baratos e cada vez mais capazes, como também de polímeros disponíveis para utilização na impressão 3D. Desde os polímeros biodegradáveis e não muito resistentes, feitos à base de material orgânico, até aos polímeros mais resistentes como o caso do ABS (acrilonitrila butadieno estireno), amplamente conhecido pela sua elevada resistência.

Ao longo do artigo apresentaram-se alguns modelos e suas vantagens e desvantagens, havendo muito mais modelos disponíveis, desde modelos de nível indústrial até modelos destinados aos makers, como o caso das famosas impressoras da série Pursa. No entanto saia completamente do âmbito deste artigo uma exploração mais exaustiva de todos os modelos de impressora e peças intermutáveis opcionais disponíveis, como o caso das extrusoras. Deixo aos leitores o desafio de aprofundarem os vossos conhecimentos neste tema caso seja do vosso interesse.

AUTOR

Escrito por Patrick Correia de Freitas

Técnico de Design de Interiores e Exteriores, está a terminar a Licenciatura em Design de Interiores e Equipamento. Procura desafiar-se diariamente na sua área e nos seus hobbies. Desenvolveu desde os seus 10anos uma enorme paixão por Gamming, na qual acabaria por se tornar um auto didacta na montagem dos seus próprios computadores de elevado desempenho gráfico. Tem um gosto particular por programas de modelação tridimensional de espaços interiores e equipamentos.

"Design is not just what it looks like and feels like. Design is how it works." (Steve Jobs)

XAMARIN PARA DOCENTES E ESTUDANTES

Era uma vez , um grupo de programadores que pensaram que deveria haver uma forma melhor de construir aplicações mobile. Assim foi criada a Xamarin: construir aplicações utilizando uma linguagem moderna e evolutiva em C#, partilha de grandes quantidades de código, construir aplicações nativas e criar a melhore experiência do utilizador. Devido ao facto de ser possível reutilizar as capacidades .NET já existentes, a Xamarin torna rápido, fácil e divertido desenvolver aplicações móveis para iOS, Android e Windows.

O desenvolvimento mobile é o mercado com o mais acentuado ritmo de crescimento dos últimos anos. O número de dispositivos móveis em utilização ultrapassa o número de habitantes do nosso planeta. A cada hora surgem novas aplicações móveis para os mais diversos efeitos, desde coisas simples como um bloco de notas a ouras mais complexas que implementam tarefas científicas e analises complexas de negócio. Além disso existe a crescente indústria do entretenimento, onde surgem frequentemente novos jogos mobile. Proporcionando experiências cada vez mais imersivas, cativantes e inovadoras,, utilizam desde realidade aumentada até georreferenciação, reconhecimento de imagem entre outras tecnologias, proporcionando ao jogador uma mistura da realidade com a realidade aumentada do jogo. Esta indústria representa uma fatia de mercado gigantesca em termos de valor financeiro, estando acessível a todos os programadores. Um simples jogo indie de plataformas pode transformar-se no próximo grande êxito dos videojogos.

A Xamarin não apenas permite criar aplicações móveis mas também auxilia a criação de jogos mobile. Frameworks como MonoGame, CocoSharp, WaveEngine, UrhoSharp, e Xenko todas suporam Xamarin, tornam muito fácil criar jogos em C#.

Recentemente as ferramentas Xamarin foram disponibilizadas no programa DreamSpark da Microsoft, colocando os programadores a apenas alguns clicks de distância de construir a próxima grande aplicação ou jogo. Ao ser completamente grátis para estudantes, a Xamarin torna possível começar a construir aplicações nativas iOS e Android no imediato e gratuitamente.

Paralelamente a esta iniciativa a Xamarin continua com o programa Xamarin Student Partner, ao qual os estudantes se podem candidatar através do formulário disponível na web, ajudando assim a espalhar o desenvolvimento mobile nativo pelos Campus em todo o mundo, bem como a manterem-se prontos a trabalhar com o desenvolvimento de aplicações mobile em C# com eventos e workshops interativos. Ao serem parte deste programa, os Student Partners ajudam as universidades a manterem-se atualizadas com as tendências mobile atuais no desenvolvimento de software. Tornar-se um student partner traz muitos benefícios, incluindo acesso a treinamento de desenvolvimento mobile interativo, ao vivo, coma Xamarin University, testes mobile com a Xamarin Test Cloud e muitos e bons recursos para ajudar na organização de Workshops.

> ***"O desenvolvimento mobile é o mercado com o mais acentuado ritmo de crescimento dos últimos anos. O número de dispositivos móveis em utilização ultrapassa o número de habitantes do nosso planeta."***

Os Xamarin Student Partners têm tudo o que precisam para começar a realizar eventos no imediato, incluindo materiais para apresentações e workshops.

Para educadores que queiram trazer o desenvolvimento mobile para as suas escolas ou universidades, podem também fazê-lo com o programa para educadores Xamarin, que auxilia os educadores a lecionarem cursos relacionados com o desenvolvimento mobile. O programa para educadores oferece as mesmas ferramentas de desenvolvimento disponíveis para os estudantes, bem como materiais de ensino e horas de Xamarin Test Cloud, simplesmente alcançáveis através do e-mail education@xamarin.com. Os Educadores que forem qualificados recebem tudo o que precisam para começar a ensinar desenvolvimento mobile em C# com Xamarin e Visual Studio, incluindo subscrições e acesso a materiais de ensino.

XAMARIN PARA DOCENTES E ESTUDANTES

Uma das grandes vantagens para os educadores recai no facto de que, ao não terem que ensinar um novo paradigma de programação, podem usar o paradigma já ensinado de programação orientada a objetos, de forma a proporcionar aos estudantes uma nova e imersiva experiência no desenvolvimento cross-platform para plataformas mobile. Ao utilizarem a linguagem de programação C# e uma ferramenta que está em expansão, dotam os futuros profissionais das tecnologias de informação de um conjunto de capacidades e valioso conhecimento, para emprego futuro, cativando o seu interesse e mantendo alta motivação de aprendizagem. A necessidade de programadores mobile continua a crescer. Em 2013 existiam 2.3 milhões de programadores mobile, sendo o seu salário médio de 71.432 USD, podendo por vezes chegar aos 110.000 USD (Economic Journal).

Em conclusão, o futuro da tecnologia passa inevitavelmente pelo desenvolvimento mobile, observando as metodologias de Mobile First. Nos dias de hoje podemos considerar os conhecimentos de desenvolvimento mobile o equivalente a saber os comandos de uma shell para se programar um computador. A avidez da indústria por aplicações móveis multi-plataforma e por profissionais competentes nestas matérias e com este tipo de skillset, cresce de dia para dia.

Xamarin é mais que uma ferramenta. É todo um conjunto de ferramentas e serviços que permitem o desenvolvimento de aplicações cross-platform, mantendo o look and feel nativo de cada plataforma com uma base de código comum. Ao aprenderem Xamarin, os estudantes estão não só a acrescentar uma mais valia a o seu skillset, mas também obtendo capacidades e conhecimentos extremamente valorizados no mercado de trabalho, ganhando portfolio que poderão apresentar de futuro.

Os educadores , ao utilizarem Xamarin e ensinarem Xamarin aos seus educandos, estão a possibilitar uma aprendizagem de um conjunto extremamente útil de capacidades e conhecimentos, proporcionando-lhes uma mais enriquecedora e completa educação. Desta forma preparam melhor os estudantes para a sua vida futura enquanto profissionais de tecnologias de informação, seguindo assim a premissa base das funções de um educador: formar as pessoas para um mundo melhor.

> ***"Uma das grandes vantagens para os educadores recai no facto de que, ao não terem que ensinar um novo paradigma de programação, podem enquadrar estas ferramentas e linguagem na matéria de já ensinada de programação orientada a objectos(...)"***

Citando um dos famosos cientistas da História da humanidade, "A melhor maneira de aprender é ensinar" - Openheimer.

AUTOR

Escrito por António C. Santos

Com uma enorme paixão por tecnologia, autodidacta desde tenra idade, cresceu com o ZX Spectrum. Tem vasta experiência em implementação e integração de sistemas ERP, CRM, ERM, BI e desenvolvimento de software por medida nas mais diversas linguagens. Diplomado do Curso de Especialização Tecnológica em Tecnologias e Programação de Sistemas de Informação pela ESTG-IPVC. Membro da Comunidade Portugal-a-Programar desde Agosto de 2007, é também membro da Sahana Software Foundation, onde é Programador Voluntário. Neste momento é aluno no Instituto Politécnico de Viana do Castelo, na Escola Superior de Tecnologia e Gestão no curso de Licenciatura em Engenharia Informática e Xamarin Student Partner nesta mesma escola Twitter:@apocsantos

HIGH TECH FASHION

Roupa. Muitos de vós podem pensar que não tem qualquer relevância, é algo que se veste, que na sua forma utilitária serve para proteger a nossa pele. Roupa é apenas roupa, não pode ter esse nível de interesse, a menos que seja para nos tornar, de certa forma, mais atraentes.

Escrever um artigo sobre moda high tech numa revista em que o público alvo é amante de tecnologia, talvez seja algo arriscado. Mas vamos imaginar que baseamos a nossa escolha de roupa num algoritmo simples, daqueles que aprendemos nas primeiras disciplinas de programação.

Se saímos à rua:

⇒ Roupa_de_Rua == TRUE && Roupa_de_Casa==FALSE;

Se ficamos em casa:

⇒ Roupa_de_Rua == FALSE && Roupa_de_Casa==TRUE;

Apostaria sem reservas que muitos dos leitores se viram neste pseudo-código.

Mas e se, a roupa se mexesse? Ou se tivesse vontade própria e mudasse, digamos, com apenas um botão? Loucura, certo? Não, já não! Um pouco de inteligência artificial talvez.

A era da tecnologia veio sem dúvida para ficar. Podemos ver tecnologia em tudo o que fazemos hoje em dia, convivemos com ela e provavelmente sem ela o mundo já não saberia como reagir a muitas situações. Exemplos bem conhecidos são os smartwatches que se tornaram gadgets bastante apetecíveis. Mas voltemos caro leitor, ao assunto principal, a roupa. Desde há milénios que os costureiros, os designers e todo o tipo de engenheiros tentam que esta roupa que nos acompanha diariamente tenha algo de diferente, algo que nos distanciei uns dos outros e nos adicione personalidade. Atualmente, o que precisamos é mesmo de uma ligação à internet e de algo que interaja connosco, algo que nos faça sentir "na nossa pele". Há alguns meses, em conversa com uma simpática rapariga informática, ela perguntava-me "achas que tens algum conselho de moda para mim? Às vezes sei que me devia vestir melhor, mas… Gostava de algo que se parecesse comigo, eu gosto de tecnologia". Hoje tenho a resposta para ti.

Comecemos pelo mais básico, Hussein Chalayan. Decorria o ano de 2000 quando este designer de moda turco teve a ideia mais brilhante da sua carreira, o que lhe valeu também variados prémios, e claro, um estupendo reconhecimento mundial. Chalayan desenvolveu em conjunto com a *B Consultants,* uma empresa londrina de engenheiros arquitetónicos, peças de vestuário que consistiam unicamente em telas brancas. A partir destas telas e da tecnologia associada a elas, Chalayan e a *B Consultants* conseguiram projetar aquilo que é hoje o primeiro vestido com imagens. Esta peça, de fundo branco, foi criada a partir de um programa de computador que permitiu aos designers elaborar uma gama de perspetivas tridimensionais, e então depois estas perspetivas foram transferidas para seda e tecidos de algodão (tecidos que são fortes condutores de energia), usando depois um processo de impressão mecanizada. Então, já te imaginaste agora a ir para o trabalho, carregares num simples botão e a tua roupa projetar imagens? Penso que seria interessante, mas talvez ainda um pouco arcaico. Ainda proveniente das mãos deste designer, surgem vários vestidos que podem ser acionados por controlo remoto. A partir da sua colaboração com estes engenheiros arquitetónicos, Chalayan desenvolve por fim vestidos que podem mudar a sua forma. Resumidamente, imaginando que vamos trabalhar com um vestido durante o dia e que durante a noite temos um evento importante ao qual temos de ir, é simples. O vestido está projetado para alterar a sua forma consoante a situação em que é necessário, o que lhe confere uma versatilidade única. Mudar de vestido é agora tão simples como clicar num botão.

Ainda um assunto pouco interessante? Um pouco menos básico, e na minha opinião, um trabalho de excelência, STUDIO XO. Vestidos voadores? O STUDIO XO consegue. Esta marca é composta pela designer Nancy Tilbury, diretora criativa da SXO, e por Benjamin Sales, diretor tecnológico da SXO. Situado em Londres, este estúdio leva a moda mais além. Estes criativos não pensam apenas no funcional ou no estético que de facto deve existir na roupa, mas pensam sim na roupa como uma segunda camada de pele – os seus estudos são maioritariamente focados no que a roupa deve servir, ao ser humano, mas também à tecnologia. Ao logo dos anos foram criando vários tipos de materiais e até minúsculos filamentos e partículas que desenham a roupa no nosso corpo, dando-nos a sensação de liberdade que por vezes tanto queremos que aquelas calças nos possam dar. Mas, calças? Não, isso é algo que não existe no dicionário da STUDIO XO. A primeira peça com impacto que Nancy criou, em conjunto com a *TechHaus*, foi um vestido chamado "Anemone 2.0", um vestido branco impresso em impressora 3D, totalmente em forma de bolha, mas o que lhe conferiu o seu sucesso, foi ser um dos maiores vestidos até à atualidade impressos em 3D, as maiores impressoras 3D do mundo. A criatividade da STUDIO XO continuou em larga escala, apresentando-nos depois a "Bubelle" – o vestido que muda de cor consoante o humor de quem o emprega. Mais interes-

sante? O vestido tem uma forma redonda, também sido este impresso. E como se junta a tecnologia de ponta a isto? Benjamin Males é o criativo de serviço nesse setor - o que Nancy desenha, Benjamin consegue dar vida, o que faz desta parceria uma das mais ricas em todo o mundo da moda. Todos estes híbridos não passaram despercebidos aos olhos da rainha do excêntrico, *Lady Gaga*. Sendo a *TechHaus* uma divisão técnica da tão aclamada *House of Gaga*, e em conjunto com a STUDIO XO, foi criado especialmente para *Lady Gaga* um vestido que voa – Volantis. Este vestido, feito em fibra de carbono e movido por seis grandes tubos de baterias, sendo todo ele ecológico (ao estilo do tão conhecido drone voador) – e é aqui que os detalhes se tornam fundamentais, pois se o vestido não fosse construído em fibra de carbono, nunca levantaria do chão. Admitamos algo que esta equipa tem de maravilhoso é um extremo conhecimento técnico. Ainda assim, e se estiveres na rua e por acaso alguém decidir que te vai atacar?

Também existe uma resposta para isso! Devo dizer que sempre adorei o filme "*Robocop*", mas a ideia de realmente trazer o seu equipamento vestido, nem sempre será a mais indicada para um dia de trabalho. Mas, e se existisse um vestido que é de facto um robot, mas com uma bonita aparência, e que alem disso, pode proteger-nos? Foi exatamente nisto que a designer Anouk Wipprecht pensou em conjunto com a Intel. Sim, a Intel faz roupa. "*Robotic Spider Dress Powered By Intel Smart Wearable Technology*", é assim que devemos procurar esta magnífica criação. Esta peça foi desenvolvida pela designer Anouk em conjunto com o arquiteto Philip H. Wilck (Estúdio Palermo) e com o grupo *New Devices* da Intel. Sim, da Intel. Eis uma palavra que os informáticos e entusiastas da tecnologia conhecem bem. A intel aliou-se a este projecto, e o vestido é "dono" de um módulo Intel Edison, o que permitiu avançar o projecto para uma versão mais madura, assim como um melhor apuramento a nível da mecatrônica e extra-sensorial. O objetivo principal era fundir a moda com a tecnologia e foi conseguido! – um vestido totalmente impresso em impressora 3D, adornado com um colar de pernas de aranha robóticas e que consegue ler pensamentos. É algo que provoca no ser humano um certo desdém, algo que consiga saber exatamente os seus pensamentos, mas neste caso, torna-se uma peça aliada à sobrevivência. O *Robotic Spider* atua como uma interface entre o corpo humano e o mundo ao seu redor – é composto por membros animatrónicos, usando sinais biométricos sem fios e as suas pernas movem-se através de sensores tecnológicos, o que lhe permite ser autónomo e aos mesmo tempo adaptável às emoções e desejos do seu portador. Este vestido é capaz de identificar 12 estados de comportamento humano, o que o faz ser agressivo ou sensível. Digamos que estamos numa festa e somos abordadas por alguém que não nos inspira confiança – imediatamente o *Robotic Spider* processa isso e movimenta o seu colar de pernas num sistema de proteger o seu "hóspede". O mesmo acontece quando temos sensação de conforto – quando queremos dançar ou abraçar uma pessoa que nos é querida, automaticamente o vestido desativa os mecanismos de defesa e deixa que a outra pessoa esteja perto de nós. A moda aliada a nós, a algo tão sensível como a nossa mente.

Então, achas que já tens roupa de trabalho?

Links onde se podem ver os vestidos a interagir:

Hussein Chalayan - https://www.youtube.com/watch?v=Ae81FcczsI8

STUDIO XO - https://www.youtube.com/watch?v=ZVtURELhy1w

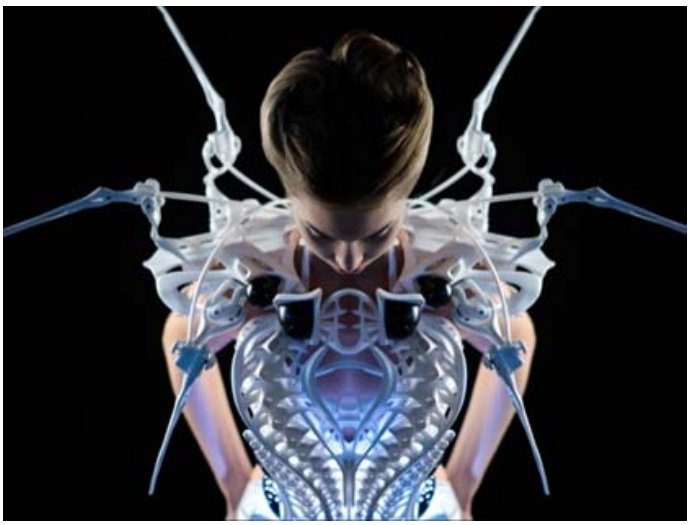
Anouk Wipprecht - https://www.youtube.com/watch?v=D40n_oZW5Ig

Fotos dos vestidos:

HUSSEIN CHALAYAN

STUDIO XO

Bubelle Dress



Volantis Flying Dress

Robotic Spider Dress

ANOUK WIPPRECHT

“Poder sem sabedoria é energia desperdiçada” (KAMILAH WILLACY)

AUTOR

Escrito por Filipa Antunes Peres.

Licenciada em Design de Moda e Têxtil e Pós-graduada em Marketing. Desde pequena que desenvolveu um forte sentido para tudo o que tenha a ver com Design.

PROJECTO EM DESTAQUE NA COMUNIDADE P@P: PORTUGOL +

Este projecto é um interpretador de uma linguagem de programação bastante simples, com uma sintaxe semelhante ao português, que permite uma fácil aprendizagem de algoritmia e programação.

De entre as muitas funcionalidades já implementadas, destacam-se as seguintes:

- comentários em linha

```
// comentário
```

- criação de variáveis com os tipos de dados : booleano, inteiro, real, texto

```
var1 : boolean
var1 : inteiro
var1 : real
var1 : texto <- "HappyHippyHippo : "
```

- estruturas de controlo : se, para, enquanto

```
var1 <- inteiro
para var de 10 até 1 então
    se var1 > 5 então
        var1 <- var1 + 2 * var
    senão
        var1 <- var1 + var
    fim de se
fim de para
```

- importação de ficheiros extra com código

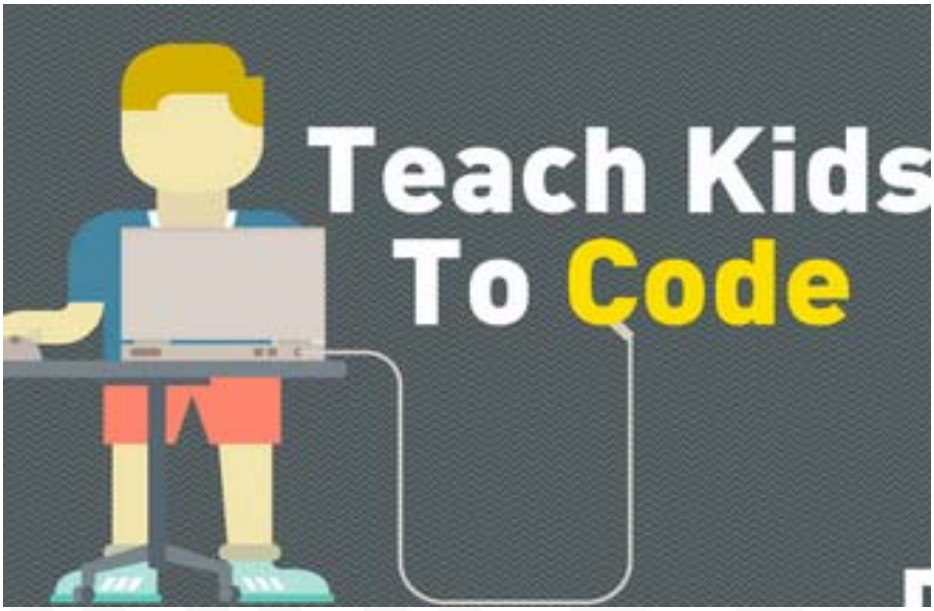
```
importar "codigo.portugol"
```

No exemplo seguinte, da autoria do criador do Portugol+ pode ver-se uma função recursiva do cálculo do factorial de um número. É de observar a simplicidade do programa e as semelhanças entre a linguagem de programação e a língua portuguesa.

```
/*
Função recursiva de calculo do factorial de um
número
> parametro[real] : n
    valor do qual o factorial irá ser calculado

> retorna[real]
    o factorial do parâmetro da função
*/

factorial : função(real n) retorna real
    // termo de paragem da função recursiva
    // - se o factorial for menor ou igual a 1 en-
tão o valor do factorial é 1
    se n <= 1 então
        retorna 1
    fim de se

    // retornar a multiplicação do valor base do
factorial com o resultado do calculo
    // do factorial desse valor base menos 1
    //
    // n! = n * (n - 1)!
    retorna n * factorial(n - 1)
fim de função
```

Pela simplicidade e pelo roadmap ambicioso, iniciativa e empenho, este projecto merece-nos a atenção! Porque tal como dizia um velho publicitário “o que é nacional é bom”, este projecto é nacional, é um bom projecto e nesta edição, ganha o nosso destaque!

Código fonte: https://github.com/HappyHippyHippo/portugol
Download executável: https://dl.dropboxusercontent.com/u/12899799/portugol/portugol.zip



Elege o melhor artigo desta edição

Revista PROGRAMAR

http://bit.do/ProgramarED51_V

Veja também as edições anteriores da Revista PROGRAMAR

50ª Edição - Setembro 2015

49ª Edição - Junho 2015

48ª Edição - Março 2015

47ª Edição - Dezembro 2014

46ª Edição - Setembro 2014

45ª Edição - Maio 2014

e muito mais em ...

www.revista-programar.info

DUVIDAS?

IDEIAS?

AJUDAS?

PROJECTOS?

portugal-a-programar
.org